



ESTIMACIÓN DE VOLUMEN SISTÓLICO Y
GASTO CARDÍACO MEDIANTE LA
IMPLEMENTACIÓN DE UN
PULSIOXÍMETRO USANDO ANÁLISIS DE
CONTORNO DE PULSO

CARLOS ALBERTO BORRÁS BAUTISTA
CARLOS AUGUSTO CALDERÓN AGUILAR

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES

BUCARAMANGA

2012



ESTIMACIÓN DE VOLUMEN SISTÓLICO Y
GASTO CARDÍACO MEDIANTE LA
IMPLEMENTACIÓN DE UN PULSIOXÍMETRO
USANDO ANÁLISIS DE CONTORNO DE PULSO

Carlos Alberto Borrás Bautista
Carlos Augusto Calderón Aguilar

Trabajo de grado presentado para obtener el título de
INGENIERO ELECTRÓNICO

DIRECTOR:

Msc. CARLOS ANDRES NIÑO NIÑO

CO-DIRECTOR:

PHD. DANIEL ALFONSO SIERRA BUENO

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES

BUCARAMANGA

2012

*Primero que todo a nuestro gran Dios y Salvador Jesucristo
a quien se lo debo todo y quien ha sido mi sustento
y fuerza para continuar siempre.*

*A mi padre y madre por todo el amor y comprensión
brindado durante estos largos años.*

*A mi familia que han sido un gran apoyo
y parte de la motivación para continuar con esta carrera.*

*A mis amigos y hermanos de la Iglesia Pentecostal Unida de Colombia
por su apoyo espiritual; muchos de ellos
personas de gran valor en mi vida.*

Carlos Alberto Borrás Bautista

*A mi madre por educarme y forjarme como soy. Por su comprensión,
cariño y constante compañía en esta etapa de mi vida.*

*A mis hermanos, y toda mi familia en general,
por su apoyo e incondicional cariño.*

A mi tío José que me tendió su mano en un momento crucial.

Mirando al cielo te digo de corazón “muchas gracias Tío!”

*A mis doñas Emilí, Luz y Leo por haberme brindado su
apoyo y acogerme en su hogar como un hijo más.*

A todos mis amigos que me ayudaron en las dificultades y compartieron mis alegrías.

*A esa fuerza invisible que aviva el espíritu,
conserva el orden y me guía por el buen camino; Dios.*

Carlos Augusto Calderón Aguilar

Agradecimientos

A nuestros directores Carlos Andrés Niño y Daniel Alfonso Sierra por ser los mejores guías en este difícil pero enriquecedor camino.

Al Dr. Carlos Arturo Conde por su valioso apoyo académico, logístico y la motivación por la investigación.

A la Fundación Cardiovascular de Colombia FCV, especialmente al Dr. Angel Maria Chavez por su calidez humana, colaboración e interés en el proyecto. A la jefe Laura y todo el personal del área en métodos no invasivos.

A todas y cada una de las personas que voluntariamente fueron monitorizadas durante las pruebas.

A la Universidad Industrial de Santander y sus profesores que cada día se preocupan en formar excelentes profesionales.

Índice general

	Página
1. Introducción	20
1.1. Pulsioximetría	21
1.1.1. Principio de funcionamiento	21
1.1.2. Onda pletismográfica	23
1.1.3. Aplicaciones	25
1.2. Electrocardiografía	26
1.2.1. Principio de funcionamiento	27
1.2.2. Aplicaciones	32
1.3. Presión arterial	33
1.3.1. Medición	35
1.4. Volumen sistólico y gasto cardiaco	37
1.4.1. Técnicas de Medición	38
2. Desarrollo del Prototipo	42
2.1. Hardware	43
2.1.1. Modulo de pulsioximetría	43
2.1.2. Electrocardiógrafo	45
2.1.3. Microcontrolador	47
2.1.4. Diseño de la tarjeta principal	50
2.1.5. Presentación final	53

2.2. Software	54
2.2.1. Filtrado	55
2.2.2. Detección de picos	58
2.2.3. Detección de punto de ascenso y onda dicrótica	62
2.2.4. Frecuencia cardíaca	63
2.2.5. Calculo del PWTT	64
2.2.6. Validación	64
2.2.7. Interfaz gráfica con Matlab	68
3. Metodología y análisis de datos	71
3.1. Tiempo de transito de la onda de pulso	71
3.2. Análisis de onda de pulso	75
3.3. Análisis de datos y resultados	78
4. Conclusiones y observaciones	87
4.1. Recomendaciones para trabajos futuros	89
Bibliografía	90
Apéndices	92

Índice de figuras

1-1. Curva de absorción de luz de la oxihemoglobina HbO_2 y la hemoglobina reducida Hb	22
1-2. Sensor de pulsioximetría en el dedo. Ubicación de los leds.	23
1-3. Onda de pulso. Amplitudes y tiempos característicos	24
1-4. Diferentes clases de onda de pulso en el dedo con incrementos en la edad o afecciones vasculares.	25
1-5. Estructura del corazón y trayecto seguido por la sangre a través de las válvulas y cavidades cardíacas	27
1-6. Electrocardiograma normal para un ciclo cardíaco	28
1-7. Acontecimientos del ciclo cardíaco. Presión auricular, presión ventricular, presión aórtica, volumen ventricular, electrocardiograma y fonocardiograma.[1, p. 107]	29
1-8. Posición de electrodos para las derivaciones estándar. Triangulo de Eindhoven. [1, p. 127]	31
1-9. Trazo de las derivaciones I, II y III	32
1-10. Relaciones entre flujo, presión y resistencia en los vasos. [2, p. 244] . . .	34
1-11. Curva de presión arterial. [2, p. 246]	35
1-12. Medición de la presión arterial con esfigmomanómetro. [2, p. 248] . . .	36
1-13. Índice cardíaco a diferentes edades. [1, p. 233]	38
1-14. a) Medida del diámetro de la válvula aórtica. b) Calculo de la integral velocidad-tiempo (VTI) en el ventrículo izquierdo.[3, p. 29]	40

2-1. Montaje del equipo	43
2-2. Modulo de pulsioximetría <i>GOLDWEI SP100 evaluation kit</i>	44
2-3. Amplificador ECG con circuito de pierna derecha ¹	46
2-4. Amplificación y filtrado para el electrocardiógrafo	47
2-5. Familia de microcontroladores MCF51QE Freescale	48
2-6. Medición de la frecuencia de señal en el LED	49
2-7. Frecuencia de aparición de valores de byte en 100000 bytes recolectados	51
2-8. Representación del envío de un paquete por el microcontrolador hacia el PC	51
2-9. Esquemático del procesado y comunicación	52
2-10. Pruebas realizadas a la tarjeta en el laboratorio	53
2-11. Visualización de las ondas PPG y ECG en el osciloscopio	53
2-12. Esfigmomanómetro y cable ECG tipo holter	53
2-13. Presentación final del equipo de medición	54
2-14. Espectro en frecuencia de la señal de ECG y el complejo QRS [4]	56
2-15. Filtrado pasa-bajas de la señal ECG. a)Sin filtrado b)Señal Filtrada	56
2-16. Filtrado pasa-altas de la señal ECG. En el tiempo a)Sin filtrado, b)Con filtrado y c)En frecuencia con filtrado	57
2-17. Filtrado pasa-bajas de la señal PPG. En el tiempo a)Sin filtrado (picos esporádicos resaltados), b)Con filtrado y c)En frecuencia con filtrado	58
2-18. Algoritmo de detección de picos	60
2-19. Visualización de umbral superior (color rojo) para detección de picos en la señal de ECG	61
2-20. Visualización de umbrales superior e inferior (rojos) para detección de picos en la señal de PPG	61
2-21. Muestra de picos detectados por algoritmo para una señal estable. De- tectados 174 de 174 picos (error = 0 %) para una señal de 76500 muestras (duración: 160 segundos)	61

2-22. Muestra de picos detectados por algoritmo para una señal con ruido electromiográfico y cambios bruscos de nivel. Detectados 207 de 205 picos (0 ignorados, 2 duplicados; error de 0,98 %), para una señal de 79500 muestras (duración: 165 segundos)	62
2-23. Puntos dicróticos detectados en una onda PPG típica	63
2-24. Determinación del tiempo de tránsito de pulso entre las señales de ECG y PPG.	64
2-25. Equipo de validación Powerlab 16/30	65
2-26. Fragmento de señales adquiridas durante la validación	65
2-27. Tiempo entre picos (en segundos) para las señales adquiridas con el equipo PowerLab 16/30 (<i>diamantes azules</i>) y el dispositivo de proyecto (<i>círculos rojos</i>) para una muestra de 111,7 segundos (147 picos) de ambas señales	66
2-28. Comparación de ritmo cardiaco de acuerdo a los tiempos entre picos mostrados en la Figura 2-27, para cada equipo	67
2-29. *Izquierda. Ondas ECG y PPG de equipo PowerLab 16/30. *Derecha. Comparación ondas ECG y PPG del dispositivo	68
2-30. Gráfico de Bland Altman para comparación de tiempos de transito de pulso entre los dos equipos	68
2-31. Interfaz gráfica del equipo de estimación de volumen sistólico y gasto cardiaco	70
3-1. Componentes del tiempo de tránsito entre la onda ECG y la onda de pulso. [5]	72
3-2. Comportamiento de la onda de presión arterial frente a la onda pletismográfica [6].	76
3-3. Diferentes formas de onda con el mismo valor de constante $K = 0,35$. [7]	77
3-4. División de la onda de pulso por medio de la onda dicrótica [7]	78
3-5. Diferentes formas de onda variando K, K_1 y K_2 [7]	78

3-6. Correlación entre el BSA (área de superficie corporal) y la constante K en el método de PWTT	79
3-7. Sesgo y precisión entre gasto cardíaco por método 1 y ecocardiografía. a) Gráfica de Bland y Altman. b) Correlación lineal	82
3-8. Sesgo y precisión entre gasto cardíaco por método 2 y ecocardiografía. a) Gráfica de Bland y Altman. b) Correlación lineal	82
3-9. Sesgo y precisión entre volumen sistólico por método 1 y ecocardiografía. a) Gráfica de Bland y Altman. b) Correlación lineal	83
3-10. Sesgo y precisión entre volumen sistólico por método 2 y ecocardiografía. a) Gráfica de Bland y Altman. b) Correlación lineal	83
3-11. Sesgo y precisión entre gasto cardíaco mediante PWTT y ecocardiografía. a) Gráfica de Bland y Altman. b) Correlación lineal	84
3-12. Sesgo y precisión entre volumen sistólico mediante PWTT y ecocardiografía. a) Gráfica de Bland y Altman. b) Correlación lineal	84
3-13. Comparación de estabilidad método PWTT vs método de análisis de pulso. a) Sujeto 1. b) Sujeto 2	85
-1. Esquemático.	93
-2. Layout capa superior e inferior	94

Índice de tablas

2-1. Descripción del protocolo de comunicación del pulsioxímetro <i>GOLDWEI SP100</i>	45
3-1. Parámetros físicos y hemodinámicos de los sujetos de prueba. *Datos representados como media \pm desviación estándar	80
3-2. Variables medidas durante las pruebas a cada persona. PWTT(*media \pm desviación estándar), CSA (área del anillo aórtico), VTI (integral velocidad-tiempo), HR (frecuencia cardiaca), SV (volumen sistólico) y CO (gasto cardiaco)	81
3-3. Resumen de correlaciones y gráficos de Bland-Altman. CO en [L/min] y SV en [mL/latido]	85

Lista de Anexos

A. Esquemático y layout de la tarjeta	93
B. Programacion del Microcontrolador MCF51QE128	95
C. Código de Procesado y visualización en Matlab	101

RESUMEN

Título:

ESTIMACIÓN DE VOLUMEN SISTÓLICO Y GASTO CARDÍACO MEDIANTE LA IMPLEMENTACIÓN DE UN PULSIOXÍMETRO USANDO ANÁLISIS DE CONTORNO DE PULSO ¹

Autores: Carlos Augusto Calderón, Carlos Alberto Borrás. ²

Palabras clave: Pulsioximetría, Electrocardiografía, Gasto Cardíaco, Tiempo de tránsito de Pulso (PWTT)

Descripción:

El gasto cardíaco (CO) y el volumen sistólico (SV) son variables fisiológicas que reflejan la eficiencia del corazón bombeando la sangre a todos los tejidos del cuerpo. El acceso a una medida de gasto cardíaco por medios no invasivos continúa en etapa de desarrollo y entre los métodos utilizados se encuentra el análisis del contorno de pulso de la onda fotopleletismográfica (PPG) y del tiempo de tránsito de onda de pulso (PWTT) obtenidos por electrocardiografía (ECG) y pulsioximetría.

Este documento muestra cómo llegar a estimar estas medidas empezando por conocer las bases fisiológicas involucradas, seguido del desarrollo del hardware y software necesarios para adquirir de una forma aceptable los datos, llegando finalmente a dos fórmulas de estimación de volumen sistólico y gasto cardíaco, utilizando la ecocardiografía Doppler transtorácica como medio de comparación. 32 personas sin arritmias continuas ni afecciones cardíacas fueron monitorizadas, registrando datos de presión arterial, edad, estatura y peso, almacenando consigo las señales de ECG y PPG en el intervalo de duración de la prueba.

Graficas de Bland-Altman y correlaciones lineales fueron usadas en el análisis estadístico de los datos dando como resultado una aceptación considerable de los métodos, en donde el relacionado con PWTT tuvo un mejor acercamiento.

¹Trabajo de Grado

²Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería Eléctrica, Electrónica y de Telecomunicaciones. Director: Carlos Andrés Niño Niño, Co-Director: Daniel Alfonso Sierra Bueno.

ABSTRACT

Title:

STROKE VOLUME AND CARDIAC OUTPUT ESTIMATION IMPLEMENTING PULSE OXIMETRY AND USING PULSE CONTOUR ANALYSIS ¹

Authors: Carlos Augusto Calderón, Carlos Alberto Borrás. ²

Key Words: Pulse Oximetry, Electrocardiography, Cardiac Output, Pulse Wave Transit Time (PWTT)

Description:

Cardiac output (CO) and stroke volume (SV) are physiological variables that reflect the efficiency of the heart pumping blood to all tissues of the body. Access to measure cardiac output by noninvasive ways are still on development stage and two of the used methods are based on pulse contour analysis of the photoplethysmographic wave (PPG) and pulse wave transit time (PWTT) obtained by electrocardiography (ECG) and pulse oximetry.

This paper shows how to get an estimate of these measures starting to know the physiological bases involved, followed by the development of the hardware and software needed to acquire an acceptable data, finally evaluating two formulas to estimate stroke volume and cardiac output, using the transthoracic Doppler echocardiography as a means of comparison. 32 people without heart disease and continuous arrhythmias were monitored, recording blood pressure, age, height and weight data, storing with them ECG and PPG signals in the interval of test duration.

Bland-Altman and linear correlations were used in the statistical analysis of the data resulting in a considerable acceptance methods, where the related PWTT had a better approach.

¹Degree Project

²Physical-Mechanical Engineering Faculty. Electrical, Electronics and Telecommunications School. Advisor: Carlos Andrés Niño Niño, Co-Advisor: Daniel Alfonso Sierra Bueno.

Capítulo 1

Introducción

El gasto cardiaco es un factor que refleja el esfuerzo del corazón para mantener un flujo adecuado de sangre hacia los tejidos del cuerpo de acuerdo a las condiciones de cada individuo, ya sea en reposo, en estado dinámico, en intervenciones clínicas, e incluso en diferentes instantes del día. El acceso a esta importante medida ha estado dominado principalmente por métodos de índole invasiva causando causando indisposición y riesgos para el paciente, haciendo del cálculo un procedimiento lento, complicado y limitado al centro clínico u hospitalario. La búsqueda de técnicas novedosas para acceder al gasto cardiaco por medios no invasivos, preferiblemente con dispositivos médicos conocidos y de fácil uso, ha encontrado en el análisis de las ondas de presión arterial, fotopletismografía y electrocardiografía, herramientas útiles para el acceso a esta medida.

Este estudio concibe la idea de estimar el gasto cardiaco mediante análisis de la onda de pulso por medio de pulsioximetría, pero en el transcurso de la investigación se encuentra escaso material que refuerce este método, debido a la ausencia de la onda de presión arterial continua y a las limitantes que la pletismografía presenta, como la sensibilidad al movimiento, cambios en la amplitud propios del paciente y agentes externos como la temperatura [7], [8].

Es así que con el firme objetivo de estimar el gasto cardiaco y volumen sistólico se incluyen dos variables más que refuerzan relación directa con esta medida; estas variables son la onda electrocardiográfica y la presión arterial, la primera utilizada en el cálculo de tiempos de tránsito del pulso junto con la onda pletismográfica y la segunda como factor de proporcionalidad al volumen sistólico; además se incluyen datos propios de cada sujeto como estatura, edad, y peso. Un análisis estadístico sera aplicado a los datos almacenados de gasto cardiaco mediante ecocardiografía transtorácica en contraste con las variables anteriormente mencionadas, a fin de obtener una relación entre ellas y formular finalmente un estimado del gasto cardiaco.

1.1. Pulsioximetría

La oximetría de pulso es un avance en monitorización de pacientes que permite obtener el nivel de oxigenación de la sangre, puede ser controlada continuamente, es relativamente simple y no-invasiva. El objetivo principal es detectar episodios de hipoxia ¹ difícilmente apreciable en medios clínicos. Su utilización en unidades de cuidado intensivo, áreas de anestesia, quirófanos, urgencias, etc. ha llegado a ser tan común como la electrocardiografía debido a la exactitud de los datos y las pocas molestias que causa a los pacientes.

1.1.1. Principio de funcionamiento

La Pulsioximetría funciona a través de dos principios de transmisión y recepción de luz para medir la saturación arterial de oxígeno: la espectro-fotometría y la fotopleτισmografía. La primera se usa para medir el porcentaje de saturación de oxihemoglobina y la segunda para diferenciar la sangre arterial de la venosa.

El pulsioxímetro compara las características de absorción de las longitudes de onda: roja (660 nanómetros) e infrarroja (940 nanómetros) que atraviesan la sangre y otros tejidos vivos (ejemplo: dedo, lóbulo de la oreja). Dicha comparación permite medir la

¹Disminución anormal de la presión parcial de oxígeno en sangre arterial

saturación de oxígeno SpO_2 . La HbO_2 (oxihemoglobina) absorbe menos luz roja y más infrarroja mientras la Hb (hemoglobina reducida) absorbe más la luz roja (*ver Figura 1-1*). El pulsioxímetro mide habitualmente el porcentaje de oxígeno unido a la hemoglobina (HbO_2).[9, p. 245]

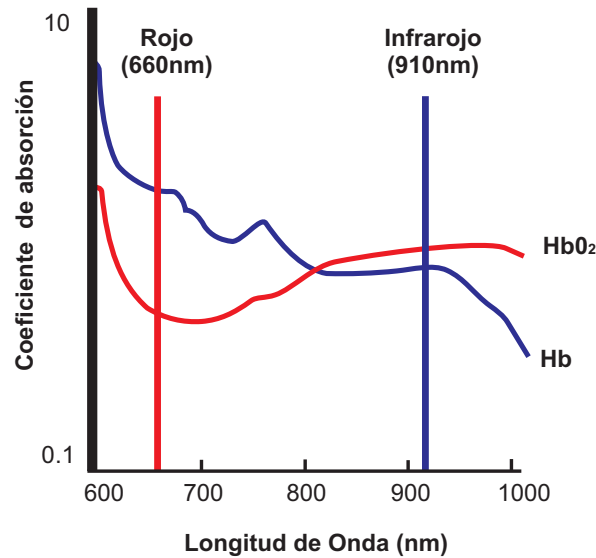


Figura 1-1: Curva de absorción de luz de la oxihemoglobina HbO_2 y la hemoglobina reducida Hb

El sensor de Pulsioximetría utiliza dos diodos emisores de luz y un foto-detector (*ver Figura 1-2*) ubicados en lados opuestos del tejido. La intensidad de luz emitida por cada diodo es conocida, luego dependiendo de la intensidad de luz que llega al foto-detector es posible determinar la intensidad de luz absorbida por la Hb . El cambio de absorción entre la diástole y la sístole se usa para determinar un estimado de la saturación arterial. Los valores de intensidad de luz son convertidos a valores de SpO_2 y onda de pulso mediante procesamiento analógico y digital.[9, p. 246]. La saturación de oxígeno es la relación entre la concentración de oxihemoglobina y la concentración de hemoglobina (oxi + desoxihemoglobina) en la sangre y se da generalmente como un valor porcentual, así:

$$\text{Saturación de Oxígeno} = \frac{[HbO_2]}{[HbO_2] + [Hb]} * 100 \quad (1-1)$$

La saturación de oxígeno puede ser evaluada por el SaO_2 o por el SpO_2 . El SaO_2 es una medida directa del contenido de oxígeno de la sangre (muestra de gas en la sangre arterial), mientras SpO_2 es una medida indirecta del contenido de oxígeno de la sangre que es detectada por pulsioximetría. La onda de pulso puede ser utilizada para calcular la frecuencia cardiaca. [9, p. 250]

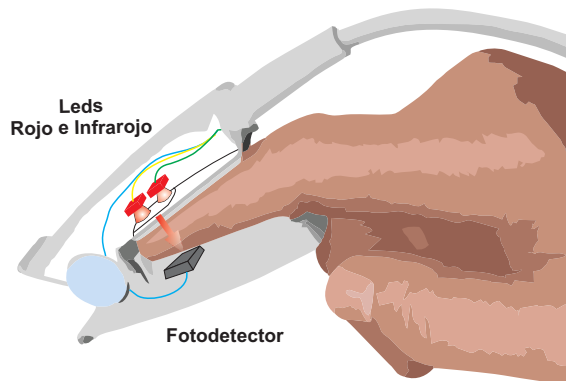


Figura 1-2: Sensor de pulsioximetría en el dedo. Ubicación de los leds.

1.1.2. Onda pletismográfica

La componente pulsátil de la onda pletismográfica, normalmente llamada componente de AC, tiene su frecuencia fundamental alrededor de 1 Hz y depende del ritmo cardiaco. Esta componente de AC esta sobrepuesta a un gran cuasi-componente DC que se relaciona con el tejido y volumen de sangre promedio. Esta componente de DC varia debido a la respiración, actividad vaso-motora y ondas vasoconstrictoras. También depende del lugar del cuerpo donde se obtenga el pulso.[10]

La apariencia del pulso fue definida en dos fases por Hertzman y Spealman: la fase anacrótica refiriéndose al flanco de subida del pulso, y la fase catacrótica refiriéndose al flanco de bajada del pulso. El pulso de presión arterial tiene similitudes con el pulso de volumen sanguíneo con cambios similares que ocurren en deficiencias vasculares tal

como amortiguación y pérdida de pulsatilidad.[10]

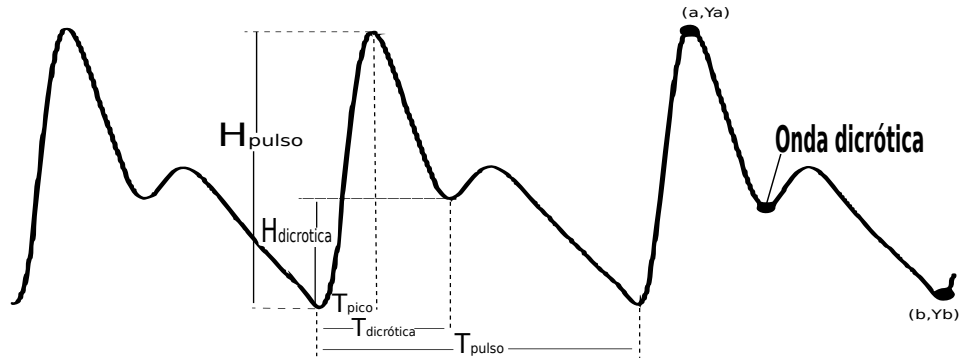


Figura 1-3: Onda de pulso. Amplitudes y tiempos característicos

La primera fase esta ligada con la sístole y la segunda con la diástole y las ondas de reflexión de la periferia. Una “muesca” (onda dicrótica) es usualmente vista en esta última fase y es más frecuente en pacientes de condiciones arteriales saludables. Parámetros relevantes como amplitud de pulso (H_{pulso}), amplitud de muesca ($H_{dicrótica}$), tiempo pico (T_{pico}), tiempo Dicrótico ($T_{dicrótica}$) y tiempo de pulso (T_{pulso}) pueden ser extraídos de la onda Pletismográfica (*figura 1-3*), y a su vez se pueden crear relaciones entre estas medidas como:

Latencia Pico = Porcentaje del tiempo pico respecto al tiempo total de pulso

$$PPL = \frac{T_{pico}}{T_{pulso}} \quad (1-2)$$

Amplitud relativa dicrótica = Porcentaje de la amplitud dicrótica a la amplitud total

$$PNRA = \frac{H_{dicrótica}}{H_{pulso}} \quad (1-3)$$

Latencia Dicrótica = Porcentaje del tiempo dicrótico respecto al tiempo total de pulso

$$PNL = \frac{T_{dicrótico}}{T_{pulso}} \quad (1-4)$$

Índice Dicrótico = Porcentaje del tiempo dicrótico respecto a la diferencia con el tiempo

total de pulso

$$NI = \frac{T_{dicrótico}}{T_{pulso} - T_{dicrótico}} \quad (1-5)$$

Por otro lado, cambios en la forma característica de la onda pletismográfica relacionados con la edad pueden dar un valioso diagnóstico acerca del sistema cardiovascular (*figura 1-4*). El objetivo de investigaciones sobre envejecimiento vascular es muy importante desde que la rigidez arterial esta asociada con hipertensión, un factor de riesgo en afecciones cerebro-vasculares y del corazón. Entre mayor sea esta rigidez más rápidamente el pulso viaja a la periferia, por lo tanto la velocidad de onda pulso se incrementa; además un aumento de la onda de presión se refleja en la rapidez de la onda reflejada.[10]

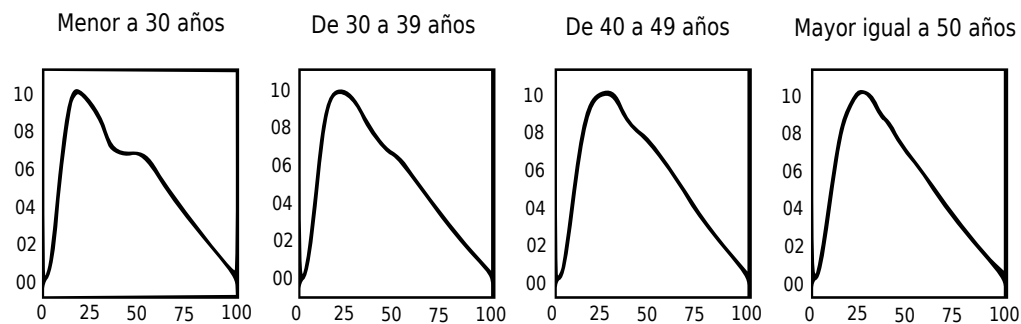


Figura 1-4: Diferentes clases de onda de pulso en el dedo con incrementos en la edad o afecciones vasculares.

1.1.3. Aplicaciones

La pulsioximetría es un excelente monitor del cambio en la oxigenación arterial, su uso en unidades de anestesia y reanimación ayuda a detectar complicaciones como la hipoxia (disminución anormal de la presión parcial de oxígeno en sangre arterial). Gracias a la pulsioximetría se han disminuido los niveles de morbilidad y mortalidad perioperatorias ². Se estima que el 80 % de los anestesistas se sienten seguros usando

²Alrededor del momento de la cirugía. Este período se extiende desde que el paciente va al hospital hasta el momento en vuelve a casa

pulsioximetría.[9, p. 248]

Se emplea también en el área de neonatología tanto en UCI (unidad de cuidados intensivos) neonatal como en sala de partos, en la monitorización inicial de pacientes con asma y brancópatas descompensados ofreciendo datos importantes del compromiso respiratorio y en el traslado de pacientes tanto intra- como extra hospitalarios.

Otro uso es el procedimiento de diagnóstico como estudios del sueño, pruebas de función respiratoria y test de ejercicios. Se destaca el uso de la onda pletismográfica en investigaciones enfocadas a la determinación no invasiva de presión arterial, volumen por latido y en relaciones con el sistema nervioso central y periférico.[11]

1.2. Electrocardiografía

La electrocardiografía consiste en el registro gráfico de la actividad eléctrica generada por el corazón. Su análisis proporciona información importante acerca del estado de este valioso órgano y permite diagnosticar enfermedades, verificar su normal funcionamiento en condiciones comunes, después de una cirugía, o simplemente para evaluar el esfuerzo físico.

El corazón es un órgano muscular contráctil formado por dos bombas separadas, una que bombea sangre hacia los pulmones y otra que bombea sangre hacia los órganos periféricos. A su vez, cada una de estas bombas es bicameral (formada por dos cámaras) pulsátil, formada por una aurícula (parte superior) y un ventrículo (parte inferior). Estas partes están detalladas en la *figura 1-5*.

Cada una de las aurículas es una bomba débil de cebado del ventrículo, que contribuye a transportar sangre hacia el ventrículo correspondiente. Los ventrículos después aportan la principal fuerza del bombeo que impulsa la sangre: hacia la circulación pulmonar por el ventrículo derecho y hacia la circulación periférica por el ventrículo izquierdo. [1, p. 103]

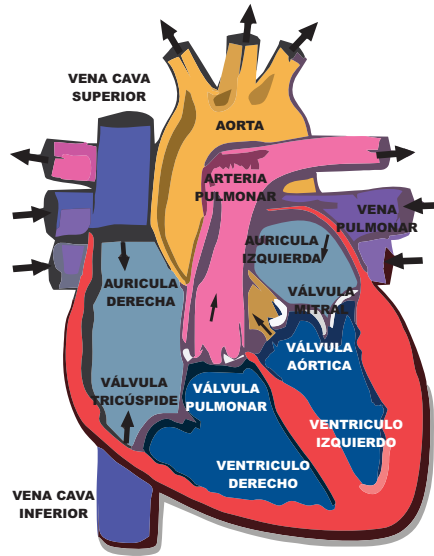


Figura 1-5: Estructura del corazón y trayecto seguido por la sangre a través de las válvulas y cavidades cardíacas

1.2.1. Principio de funcionamiento

La actividad eléctrica cardíaca en condiciones normales, es el resultado de un impulso que se origina en una célula o en un grupo de células, que constituyen el marcapasos, y que se propaga a todas las fibras de las aurículas y los ventrículos causando la debida contracción de estos y el bombeo de la sangre. Una vez que el impulso cardíaco atraviesa el corazón, la corriente eléctrica también se propaga desde el corazón hacia los tejidos adyacentes que lo rodean.

Una pequeña parte de la corriente se propaga hacia la superficie corporal; si se colocan electrodos en la piel en lados opuestos del corazón se pueden registrar los potenciales eléctricos generados durante el ciclo cardíaco; este registro se conoce como electrocardiograma (*visto en la Figura 1-6*). Los fenómenos cardíacos que se producen desde el comienzo de un latido cardíaco hasta el comienzo del siguiente se denominan ciclo cardíaco.

- Cada ciclo es iniciado por la generación espontánea de un potencial de acción en el nódulo sinusal en la aurícula derecha, cerca del orificio de la vena cava superior.

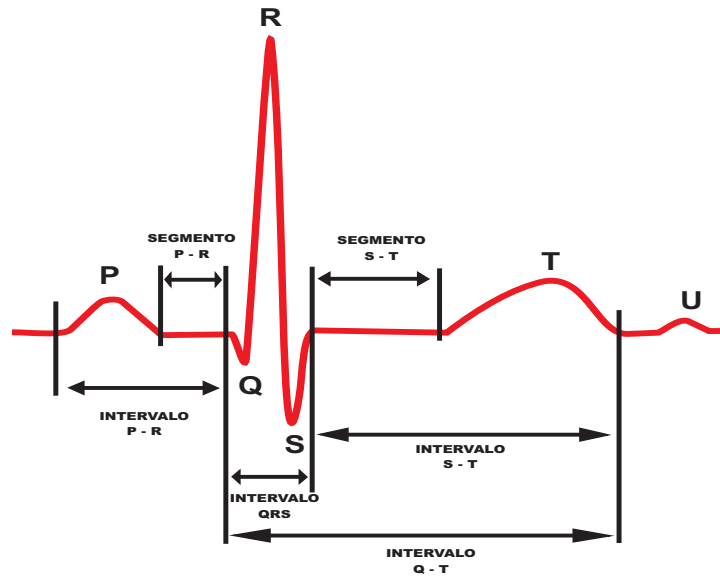


Figura 1-6: Electrocardiograma normal para un ciclo cardiaco

- El potencial de acción se desplaza a través de ambas aurículas y el nódulo y el haz de AV hacia los ventrículos.
- Hay un retraso de más de 0,1 segundos durante el paso del impulso cardíaco desde las aurículas a los ventrículos, lo que permite que las aurículas se contraigan antes que los ventrículos.
- Esta contracción permite que la sangre en las aurículas sea bombeada hacia los ventrículos antes que comience la intensa contracción ventricular.[1, p. 106]

Los ventrículos se llenan de sangre durante la diástole y se contraen durante la sístole. Las tres curvas superiores en la Figura 1-7 muestran los cambios de presión en la aorta, en el ventrículo izquierdo y en la aurícula izquierda respectivamente. Las curvas inferiores muestran los cambios en el volumen ventricular, el electrocardiograma y el fonocardiograma (registro de los ruidos cardíacos).

Cada una de estas acciones dadas en el corazón tienen una forma de onda en el registro electrocardiográfico:

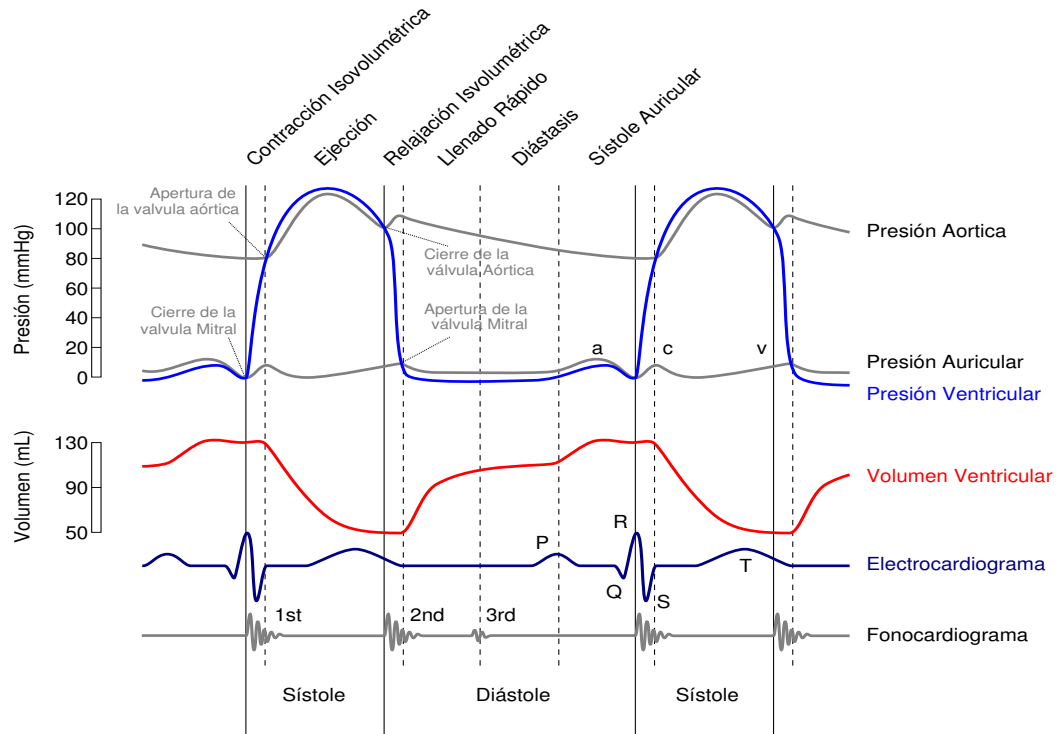


Figura 1-7: Acontecimientos del ciclo cardíaco. Presión auricular, presión ventricular, presión aórtica, volumen ventricular, electrocardiograma y fonocardiograma. [1, p. 107]

- La onda P está producida por la propagación de la despolarización en las aurículas, que causa la contracción auricular. La presión auricular aumenta inmediatamente después de la onda P.
- El complejo *QRS* aparece como consecuencia de la despolarización ventricular aproximadamente 0,16s después del inicio de la onda P y marca el comienzo de la contracción ventricular. Después, la presión ventricular comienza a aumentar.
- La onda T ventricular se debe a los potenciales que se generan cuando los ventrículos se recuperan del estado de despolarización, es decir, cuando las fibras del músculo ventricular comienzan a relajarse. Por tanto, la onda T se produce un poco antes del final de la contracción ventricular. [1, p. 107]

Voltajes normales en un electrocardiograma

Los voltajes de las ondas que se registran en el electrocardiograma normal dependen de la manera en la que se aplican los electrodos a la superficie del cuerpo y de la proximidad de los electrodos al corazón. Cuando un electrodo está colocado directamente sobre los ventrículos y un segundo electrodo está colocado en otra localización del cuerpo alejada del corazón, el voltaje del complejo QRS puede ser de hasta 3 a 4 mV. Incluso este voltaje es pequeño en comparación con el potencial de acción monofásico de 110 mV que se registra directamente en la membrana del musculo cardiaco.

Cuando los electrocardiogramas se registran con electrodos en los dos brazos o en un brazo y una pierna, el voltaje en el complejo QRS habitualmente es de 1,0 a 1,5 mV desde el punto más elevado de la onda R hasta el punto más profundo de la onda S. El voltaje de la onda P está entre 0,1 y 0,3 mV y el de la onda T está entre 0,2 y 0,3 mV. [1, p. 125]

Derivaciones electrocardiográficas

Las diferentes configuraciones de posición de los electrodos en el cuerpo son llamadas derivaciones y registran diferentes formas de onda dependiendo de la posición de los electrodos. Las derivaciones más comunes, son las llamadas derivaciones bipolares, en las cuales se posicionan los electrodos en dos extremidades, como se muestra en la Figura 1-8. Son tres las derivaciones bipolares en las extremidades, denominadas derivación I, II y III.

- Para registrar la **derivación I**, el terminal negativo del electrocardiógrafo está conectado al brazo derecho (RA) y el terminal positivo al brazo izquierdo (LA). Durante el ciclo de despolarización, el punto en el que el brazo derecho se conecta con el tórax es electronegativo respecto al punto en el que se conecta el brazo, y el electrocardiógrafo registra una señal positiva cuando se usa esta derivación.
- Para registrar la **derivación II** de las extremidades, el terminal negativo del electrocardiógrafo se conecta al brazo derecho y el terminal positivo a la pierna izquierda. Por tanto, cuando el brazo derecho es negativo respecto a la pierna

izquierda, el electrocardiógrafo registra una señal positiva.

- El registro de la **derivación III** se hace conectando el terminal negativo del electrocardiógrafo al brazo izquierdo y el terminal positivo a la pierna izquierda. Esto significa que el electrocardiógrafo registra una señal positiva cuando el brazo izquierdo es negativo respecto a la pierna izquierda.

El **triángulo de Einthoven**, dibujado en la Figura 1-8, alrededor de la zona del corazón ilustra que los dos brazos y la pierna izquierda forman vértices de un triángulo. La ley de Einthoven afirma que si en cualquier momento dado se conocen los potenciales eléctricos de dos cualesquiera de las tres derivaciones electrocardiográficas bipolares de las extremidades, se puede determinar matemáticamente la tercera simplemente sumando las dos primeras (aunque se debe tener en cuenta que se deben observar los signos positivos y negativos de las diferentes derivaciones cuando se haga esta suma).

Matemáticamente este principio, denominado ley de Einthoven, es cierto en cualquier momento dado mientras se registren los tres electrocardiogramas bipolares estándar. En la Figura 1-9 se muestra el registro electrocardiográfico de las derivaciones I, II

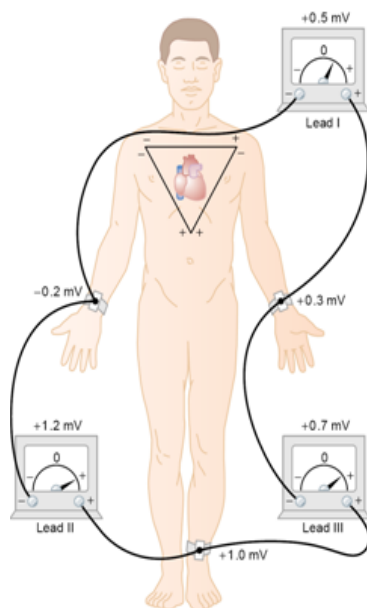


Figura 1-8: Posición de electrodos para las derivaciones estándar. Triángulo de Einthoven. [1, p. 127]

y III. Los electrocardiogramas de estas tres derivaciones son similares entre si porque todas registran ondas P positivas y ondas T positivas, y la mayor parte del complejo QRS también es positiva en todos los electrocardiogramas. Cuando se analizan los tres registros electrocardiográficos se puede demostrar que en cualquier momento dada la suma de los potenciales de las derivaciones I y III es igual al potencial de la derivación II, lo que ilustra la validez de la ley de Einthoven.[1, 128]

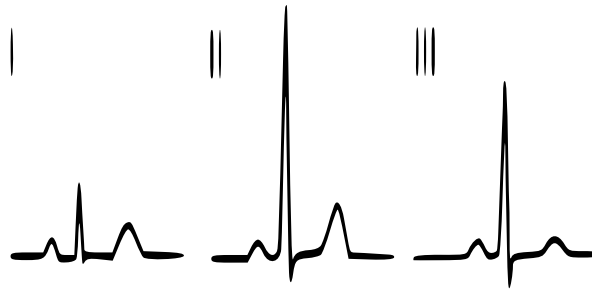


Figura 1-9: Trazo de las derivaciones I, II y III

Selección de la derivación II para el proyecto

Para la determinación correcta de los parámetros necesarios en este proyecto, el punto más determinante de la onda electrocardiográfica es el pico R del complejo QRS, por lo tanto se necesita que tenga la mayor amplitud de la onda para facilitar su correcta localización.

El pico R presenta su mayor amplitud en la derivación II, que según la ley de Einthoven corresponde a la suma de las derivaciones I y III [1, p. 128] y por tanto en un electrocardiograma común con derivaciones bipolares (picos R positivos) el pico R en la derivación II siempre será mayor que el correspondiente en las derivaciones I y III.

1.2.2. Aplicaciones

Dentro de las aplicaciones más importantes de la electrocardiografía se encuentran:

- La evaluación de la función cardiaca y la detección de arritmias y algunas enfermedades del corazón que presentan un patrón reconocido en el registro electro-

cardiográfico, algunas de ellas, bien conocidas, son la cardiopatía isquémica o la fibrilación auricular entre otras.

- Permite el seguimiento del estado de un paciente sometido a algún tratamiento o cirugía, pues es posible almacenarlo en medio físico (papel electrocardiográfico) o magnético para su futura comparación y evolución del paciente.
- La electrocardiografía también es usada en la evaluación del funcionamiento de marcapasos e implantes al corazón.
- En la determinación de la eficacia de ciertos medicamentos para el corazón, el sistema circulatorio u otros órganos relacionados con la función cardíaca.
- También es usado en la determinación del estado físico de la persona, evaluando el desempeño del corazón durante el ejercicio físico, el esfuerzo o el reposo.[11]

1.3. Presión arterial

La *presión arterial* se define como la fuerza ejercida por la sangre en forma perpendicular sobre la pared del vaso y que produce una *tensión* o distensión de la pared. La tensión que soporta el vaso, tiene una dirección tangencial a la pared y es en parte consecuencia de la presión. La presión y la tensión se relacionan por medio de la ley de Laplace:

$$T = \frac{P * r}{2e} \quad (1-6)$$

Donde r representa el radio del vaso y e el espesor de la pared vascular.

Es decir que la tensión a un determinado valor de presión es directamente proporcional al radio. Entre menor sea el radio de un vaso, menor será la tensión que soporta, aunque la presión sea la misma que en un vaso de mayor radio. Esto explica porque vasos pequeños sometidos a elevadas presiones no se rompen y en cambio un aneurisma (dilatación anormal y permanente de un vaso) aunque sufre la misma presión que el

resto, su radio aumenta y debe soportar una mayor tensión. [2, p. 242]

La *diferencia de presión* entre los dos extremos de un vaso, también denominada *gradiente de presión* es la fuerza que empuja la sangre a través del vaso y determina el flujo sanguíneo que lo atraviesa. Mientras el impedimento que el flujo sanguíneo encuentra en el vaso, se conoce como resistencia vascular (*figura1-10*). Esta relación es análoga, a la relación entre corriente, diferencia de tensión y resistencia en un circuito eléctrico expresada por la ley de Ohm, y está dada por la ecuación:

$$F = \frac{\Delta P}{R} \quad (1-7)$$

Donde F es el flujo sanguíneo, ΔP es la diferencia de presión ($P_2 - P_1$) entre los dos extremos del vaso y R es la resistencia vascular [1, p.164]

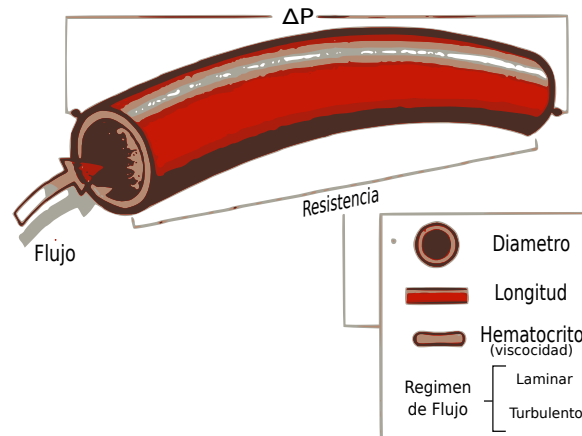


Figura 1-10: Relaciones entre flujo, presión y resistencia en los vasos. [2, p. 244]

Diferencia de presión

Como el corazón bombea sangre hacia la aorta, la presión en ella se mantiene elevada, con una media de 100 mmHg. La función de bomba del corazón es cíclica (sístole y diástole), por lo tanto la presión oscila entre un valor máximo en sístole PAS (Presión arterial sistólica) de alrededor de 120 mmHg y un valor mínimo en diástole PAD (Presión arterial diastólica) de 80 mmHg en un adulto normal. La presión arterial

sistólica depende en mayor medida del volumen sistólico (SV) y de la elasticidad de la aorta; además se ve influida en forma indirecta por la resistencia periférica (RP). La presión arterial diastólica depende sobre todo de la RP y de la distensibilidad del sistema arterial.

La *presión de pulso* es una medida de la amplitud de la onda del pulso mostrada en la *figura 1-11* y se define como al diferencia entre la PAS y la PAD ($PAS - PAD$), y depende en primer término del SV y de la distensibilidad del sistema arterial.

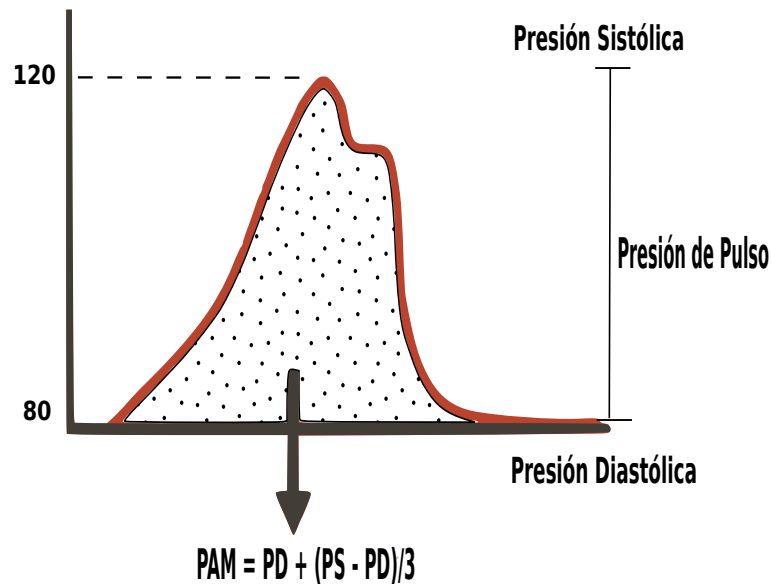


Figura 1-11: Curva de presión arterial. [2, p. 246]

Al querer hallar el valor medio de la onda de presión arterial se debe hallar el valor del área bajo la curva de presión sobre el tiempo periodo de la onda; para simplificar esta operación se ha aproximado la *presión arterial media* (PAM) con el siguiente valor:

$$PAM = PAD + \frac{PAS - PAD}{3} \quad (1-8)$$

1.3.1. Medición

El método más ampliamente usado en la determinación de la presión arterial es el auscultatorio y se realiza con un esfigmomanómetro, como se muestra en la *figura 1-12*.

Para realizar la toma de la medida se insufla³ una funda inflable de presión alrededor del brazo hasta superar el valor de la presión sistólica, impidiendo así el flujo de sangre en la arteria humeral; se coloca el estetoscopio sobre la arteria y se comienza a desinflar la funda descendiendo la presión con lentitud (aprox. 2 mmHg por segundo). El descenso de presión permitirá el paso del flujo sanguíneo cuando la presión de la funda caiga por debajo de la presión sistólica. El flujo circula ahora por un vaso de diámetro disminuido, por lo que se vuelve turbulento y se hace audible con el estetoscopio (ruido de Korotkoff), y a partir de ese momento cada latido cardíaco será audible hasta que la presión en la funda inflable, que continúa en caída, libere por completo el vaso arterial; en ese momento el flujo sanguíneo volverá a ser laminar y retornará el silencio auscultatorio. El valor registrado al escuchar el primer ruido después de empezar desinflar la funda representa el valor de la presión sistólica y el último audible el de la presión diastólica. [2, p. 247]

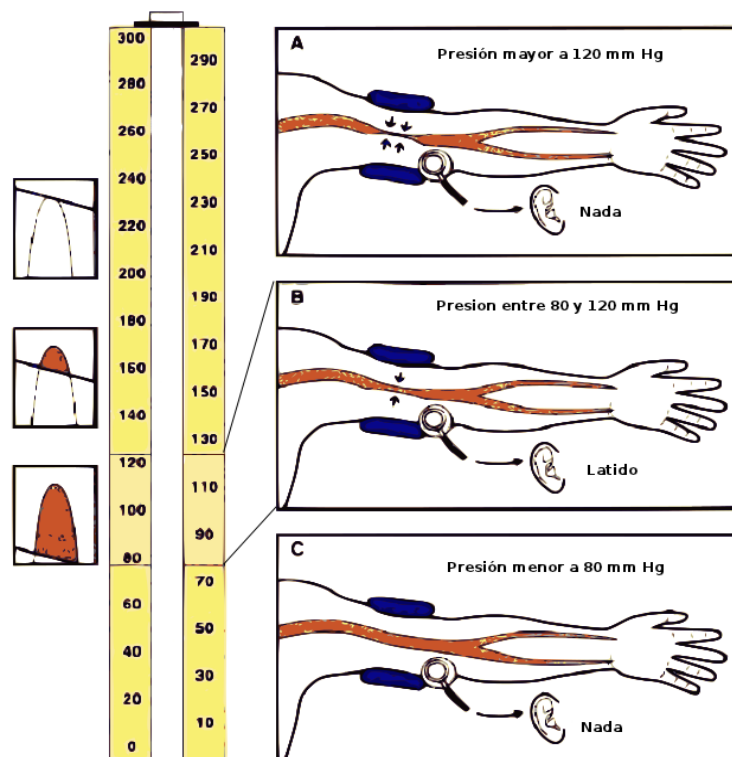


Figura 1-12: Medición de la presión arterial con esfigmomanómetro. [2, p. 248]

³Equivalente a inflar

1.4. Volumen sistólico y gasto cardiaco

La función primaria de corazón es la de impartir energía a la sangre con el fin de sostener una presión arterial sanguínea suficiente para perfundir de una manera adecuada los órganos. El corazón logra esta tarea contrayendo sus paredes musculares generando suficiente presión que expulse la sangre desde el ventrículo izquierdo, a través de la válvula aórtica hacia la aorta. Cada vez que el ventrículo izquierdo se contrae una cantidad de volumen es expulsado hacia la aorta. Esta cantidad de volumen-latido o **volumen sistólico (SV)** multiplicado por el numero de latidos por minuto (frecuencia cardiaca) se le conoce como **gasto cardíaco (CO)** (*ecuación 1-9*). [12, p. 67]

$$CO = SV * HR \quad (1-9)$$

Por lo tanto cambios en el volumen sistólico alteran el gasto cardiaco. Las unidades para gasto cardiaco son mililitros/minuto (*ml/min*) o litros/minuto (*L/min*) y para el volumen sistólico mililitros/latido (*ml/latido*). Algunas veces el gasto cardiaco es expresado como un índice cardiaco (*ecuación 1-10*), resultado de dividir el gasto cardiaco por un estimado de la superficie corporal en metros cuadrados, BSA (*body surface area*).

$$IC = \frac{CO}{BSA} \quad (1-10)$$

Existen diversas formulas para el calculo del área de superficie corporal pero las más usadas son la fórmula Mosteller (*ecuación 1-11*) y Dubois (*ecuación 1-12*).[12, p. 68]

$$BSA(m^2) = \sqrt{\frac{Estatura(cm) * Peso(Kg)}{3600}} \quad (1-11)$$

$$BSA(m^2) = 0,20247 * Estatura(m)^{0,725} * Peso(Kg)^{0,425} \quad (1-12)$$

Un rango normal para índice cardiaco es de 2,6 a 4,2 [*L/min/m²*]. En la *figura 1-13* se puede apreciar el cambio del índice cardiaco a través de la edad.

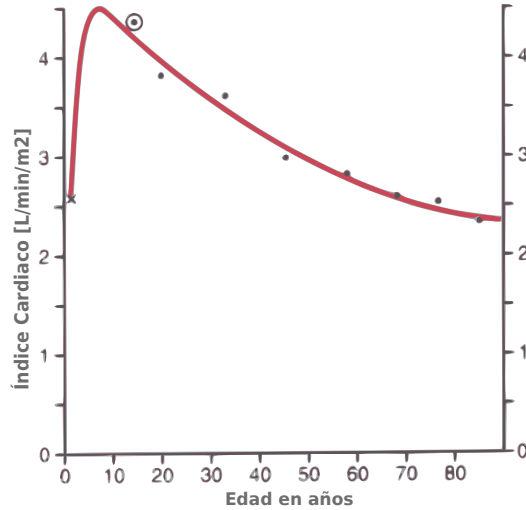


Figura 1-13: Índice cardíaco a diferentes edades. [1, p. 233]

1.4.1. Técnicas de Medición

Aunque no se ha llegado a un método “*gold standard*” en la determinación del gasto cardíaco, en la práctica clínica se cuenta con métodos ampliamente utilizados para la medición del gasto cardíaco e igualmente del volumen de eyección sistólico; más de ocho empresas ofrecen sus productos en el mercado para la medición de gasto cardíaco, pero muchos de esos dispositivos no miden el gasto en tiempo real y son invasivos.[13]

El más cercano a ser llamado un “*gold standard*” es el método de determinación de gasto cardíaco por termodilución (TCO), donde se calcula el promedio de tres valores obtenidos durante la introducción de un catéter arterial por 3 a 5 minutos. La técnica de catéter arterial derecho o pulmonar (PAC) permite la determinación del gasto cardíaco continuo después de un tiempo de 2 a 3 minutos. Ambos son invasivos y requieren de un procedimiento quirúrgico para su aplicación. Las únicas técnicas no invasivas utilizadas en la práctica médica son la bioimpedancia eléctrica torácica y la ecocardiografía transtorácica Doppler (TTE); otras técnicas semi-invasivas conocidas son la reinhalación parcial de dióxido de carbono y la ultrasonografía Doppler esofágica (ED), pero también están llenas de inconvenientes, además de no ser muy amigables con el usuario.[13]

Han surgido recientemente nuevas técnicas que miden el gasto cardiaco usando la forma de onda de presión arterial y la dilución de litio, esta última siendo semi-invasiva debido a la inyección del indicador y un electrodo sensible al ion, pero en condiciones de bajo gasto cardiaco (baja eyección de volumen de sangre por latido), los valores estimados pueden llegar al 20 % de error en comparación con la técnica de termodilución [9, p. 211]. Y es durante los periodos de bajo gasto cardiaco cuando son más deseables las medidas exactas; además en pacientes con enfermedad cardiaca valvular y *shunts intracardiacos*⁴ el valor de gasto cardiaco obtenido por las técnicas disponibles llega apenas a una aproximación del valor real.[13]

Método de referencia: Ecocardiografía transtorácica Doppler

Se escoge la ecocardiografía transtorácica Doppler por ser un método no invasivo, con menores complicaciones al paciente y al alcance gracias a su uso en la Fundación Cardiovascular de Colombia FCV. La ecocardiografía transtorácica Doppler es una variedad de la ecografía tradicional que está basada en el uso del ultrasonido y el efecto Doppler. La cantidad de flujo, o volumen sistólico, a través de un vaso o un orificio se determina como el promedio de velocidades de las células sanguíneas atravesando esa estructura multiplicado por el área de sección transversal (CSA) de la estructura donde la velocidad es medida. El CSA es calculada, asumiendo un orificio circular, midiendo el diámetro en el punto de interés.

Las medidas de flujo generalmente son tomadas en la abertura valvular, porque la medida del diámetro de la abertura valvular es menos variable a través del ciclo cardiaco. Debido a la relativa facilidad para medir la velocidad del flujo y el diámetro de la abertura, las medidas de volumen de eyección sistólico se toman en la válvula aórtica. Luego, el gasto cardiaco es calculado multiplicando el volumen sistólico con el ritmo cardiaco.[3, p. 27]

⁴Pasaje de fluido de derecha a izquierda, izquierda a derecha o bidireccional entre las cámaras del corazón

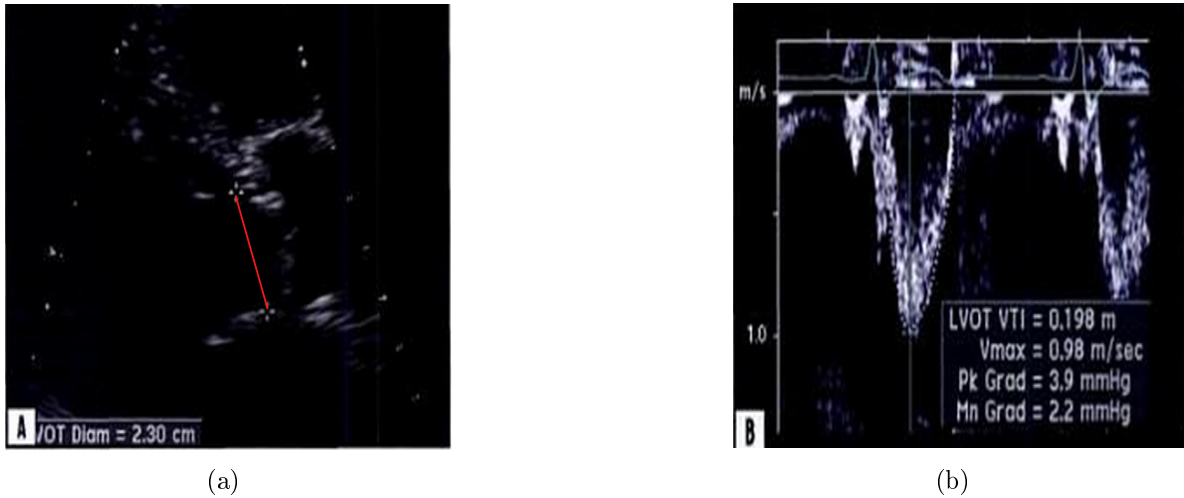


Figura 1-14: a) Medida del diámetro de la válvula aórtica. b) Cálculo de la integral velocidad-tiempo (VTI) en el ventrículo izquierdo.[3, p. 29]

Medición del diámetro valvular. El diámetro de la abertura de la válvula aórtica es calculado observando por medio del ecocardiógrafo posicionando el transductor en el tercer o cuarto espacio intercostal izquierdo del tórax, la imagen obtenida es mostrada en la Figura 1-14.

Medición del flujo aórtico. El flujo sanguíneo a través de la aorta durante la sístole debe ser observado desde el plano apical del ventrículo izquierdo así se obtendrá una deflexión negativa de rápido descenso, (figura 1-14), esto debido a que el flujo sanguíneo registrado se aleja del transductor. La medida de velocidad media requerida se obtiene calculando la integral velocidad contra tiempo (VTI) del flujo en el mismo punto donde se midió el diámetro de abertura de la válvula aórtica. La VTI es determinada por el ecocardiógrafo en unidades lineales, [cm] generalmente.

Con los datos observados en la Figura 1-14 es posible calcular el gasto cardiaco mediante el siguiente procedimiento:

- Se calcula el área de sección transversal CSA asumiendo un vaso circular

$$\begin{aligned} CSA_{LVOT} &= \pi * \left(\frac{Diametro}{2}\right)^2 \\ &= 3,14 * \left(\frac{2,3}{2}\right)^2 = 4,1 \text{ cm}^2 \end{aligned}$$

donde LVOT = tracto de salida ventricular izquierdo

- Luego se calcula el volumen sistólico como el producto CSA por la integral velocidad tiempo VTI

$$\begin{aligned} SV &= CSA_{LVOT} * VTI_{LVOT} \\ &= 4,1 \text{ cm}^2 * 20\text{cm} = 82 \text{ cm}^3 = 82 \text{ ml} \end{aligned}$$

- Finalmente se calcula el gasto cardiaco como el producto SV por frecuencia cardiaca HR

$$\begin{aligned} CO &= SV * HR \\ &= 82 * 70 [\text{lat}/\text{min}] = 6478 [\text{ml}/\text{min}] \\ &= 6,5 [L/\text{min}] \end{aligned}$$

Capítulo 2

Desarrollo del Prototipo

Anteriormente se han mencionado las variables que interactúan en este proyecto como la electrocardiografía, la pulsioximetría y la presión arterial, por lo tanto se debe contar con un hardware adecuado que permita acceder a tales variables de una forma segura y confiable. En la Figura 2-1 se puede tener una visión del equipo que realizaría la adquisición, procesado y visualización de los datos de electrocardiografía y pulsioximetría, así como el ingreso de datos inherentes al paciente y un estimado final de volumen sistólico y gasto cardiaco.

Para el desarrollo de este equipo se ha importado un Pulsioxímetro del fabricante GOLDWEI, además se ha fabricado una PCB¹ donde se incluyan la fuente de alimentación, el electrocardiógrafo y el microcontrolador. Los datos que son recibidos en el computador mediante comunicación serial vía USB, serán procesados y visualizados gracias a la herramienta computacional de Matlab. y podrá ser ejecutada en un computador portátil o de escritorio.

A continuación se hace una descripción del hardware y software desarrollado para la ejecución del presente proyecto, una información detallada del código de programación del microcontrolador y del código de Matlab puede ser encontrada en los apéndice 4.1 y 4.1 respectivamente.

¹Printed Circuit Board

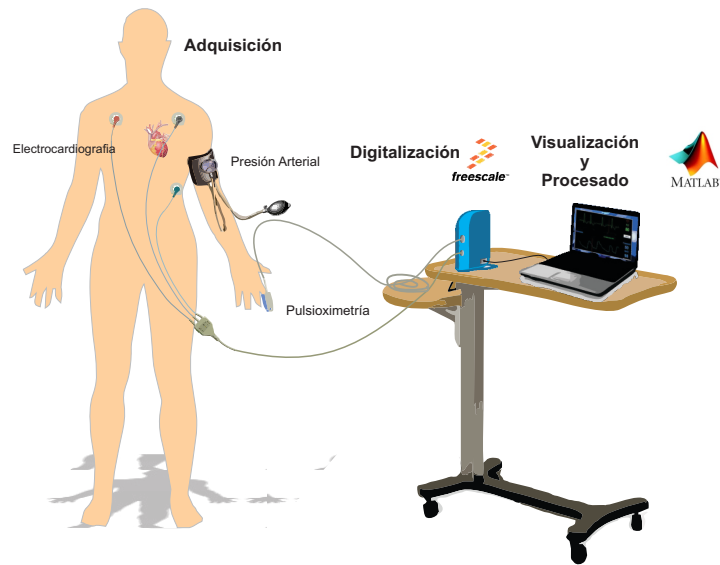


Figura 2-1: Montaje del equipo

2.1. Hardware

2.1.1. Modulo de pulsioximetría

El equipo de pulsioximetría adquirido es el módulo OEM² *SPO2 SP100 evaluation kit* de la empresa *GOLDWEI*, *Figura 2-2*.

El módulo *SpO₂* modelo SP100 es usado en la inspección o medida de la saturación de oxígeno del paciente, onda de pulso y ritmo cardiaco, puede ser usado en infantes, recién nacidos y adultos. El modulo SP100 es compatible con todos los sensores BCI, en su versión “*evaluation kit*” incluyen los conectores, el sensor y alimentación USB.

Principales características

- Pequeño y de fácil instalación.
- Procesado de señal mejorado con el fin de reducir efectos extraños asociados al

²Fabricante de equipo original, en general son dispositivos que carecen de envoltorio colorido o no incluyen software para ser comprados por otra empresa y vendidos bajo su propia marca

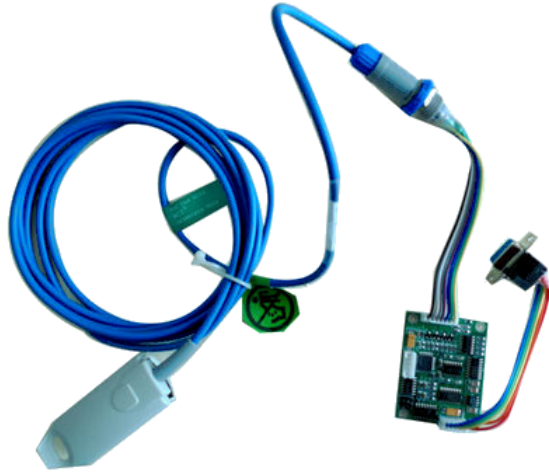


Figura 2-2: Modulo de pulsioximetría GOLDWEI SP100 evaluation kit

movimiento y baja perfusión.

- Trabaja con una fuente simple de 5V DC.
- Con pletismografía, intensidad de señal y frecuencia cardiaca.
- Protocolo de comunicación RS232.

Alimentación

El módulo SP100 de GOLDWEI funciona con $5V \pm 5\%$ DC y un consumo de 40 mA.

Medición de saturación de oxígeno

La medida funcional es de un 0-99 % SpO_2 . Tiene una precisión de $\pm 2\%$ entre 70-100 %, en menos de 70 % la precisión es indefinida.

Protocolo de comunicación serial

La comunicación serial es transmitida con voltajes a nivel TTL. La configuración del puerto es: 4800 baudios, 1 bit de start, 8 bits de datos, 1 bit de stop y paridad impar. Los datos son transferidos a una frecuencia de 60 paquetes por segundo, donde el último

bit de cada byte es de sincronización. Ver tabla 2-1

Byte	Bit	Descripción
1	0-3	Intensidad de la señal en el rango de 0-8
	4	0
	5	0
	6	Señal de Pulso =1.
	7	Bit de sincronización = 1.
2	0-6	Onda Pletismográfica
	7	Bit de sincronización = 0.
3	0-3	Barra gráfica
	4	Sensor en dedo=0, Sin sensor=1.
	5	0
	6	Bit mas significativo de frecuencia cardiaca
	7	Bit de sincronización = 0.
4	0-6	Frecuencia Cardiaca
	7	Bit de sincronización = 0.
5	0-6	Saturación de Oxígeno
	7	Bit de sincronización = 0.

Tabla 2-1: Descripción del protocolo de comunicación del pulsioxímetro GOLDWEI SP100

Como limitante en este módulo se encuentra que tiene una baja tasa de muestreo de 60 muestras por segundo, por tal motivo se extrae del modulo la señal analógica de onda pletismográfica y a través de un *buffer* se envía al microcontrolador para ser muestreada a un valor mas alto. No obstante los datos que salen en forma serial son igualmente útiles y se envían al microcontrolador.

2.1.2. Electrocardiógrafo

El desarrollo del electrocardiógrafo se hace con el fin de obtener el tiempo de transito de la onda de pulso que va desde el corazón hasta el dedo donde se coloca el pulsioxímetro, para esto se deben obtener claramente diferenciados los picos R de la señal de ECG. Para lograrlo, el dispositivo debe entregar una señal en lo posible libre de ruido, ser seguro para el usuario, económico y con bajo consumo de energía.

El diseño esta compuesto por dos etapas de amplificación junto con un circuito de pierna derecha para aumentar el rechazo en modo común *CMR* (véase en la Figura 2-3), seguido de un filtro RC pasa-altas de 0,5 Hz para eliminar la componente DC y un filtro pasa-bajas de 40 Hz para eliminar ruido de 60 Hz y alta frecuencia (véase esquemático en la Figura 2-4).

El circuito de pierna derecha toma la señal en la pierna derecha, la amplifica, invierte y realimenta a través de la resistencia de entrada. Esta realimentación negativa disminuye la señal de modo común a la entrada del amplificador.

Se seleccionó el amplificador operacional *OPA2350* para el circuito de pierna derecha, y el IC *TLC1079* para el filtrado y acople de impedancias, ambos fabricados por Texas Instruments. Como amplificador de instrumentación con precisión y baja potencia se selecciona el *INA129* de Texas Instruments, el cual posee un *CMR* de mínimo 120dB, entradas protegidas y es aconsejado para instrumentación médica.

Su ganancia está dada por:

$$G = 1 + \frac{49,4K\Omega}{R_G}$$

En la Figura 2-3 se muestra la configuración del circuito para el ECG, escogiendo una R_G de $5,4K\Omega$ para obtener una ganancia aproximada de 10 V/V. en la primera etapa y una R_G de $1K\Omega$ en la segunda etapa (figura 2-4) con ganancia aproximada de 50 V/V.

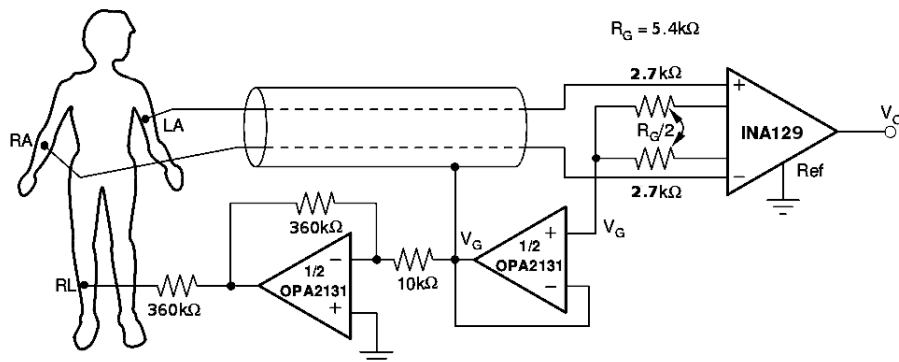


Figura 2-3: Amplificador ECG con circuito de pierna derecha¹

¹Tomado del datasheet del INA129 de Texas Instruments, Figura 8.

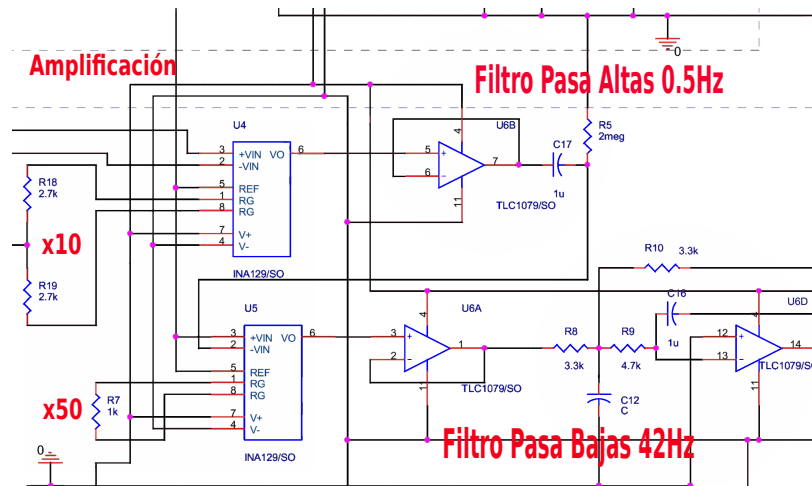


Figura 2-4: Amplificación y filtrado para el electrocardiógrafo

Los componentes utilizados en el ECG son de montaje superficial y están integrados junto con el microcontrolador y la fuente en la misma tarjeta. Para un diagrama completo de la tarjeta puede referirse al apéndice 4.1

2.1.3. Microcontrolador

La programación del microcontrolador (μC) se realizó con el software *CodeWarrior for Microcontrollers V6.3 Special Edition* de Freescale. La inicialización del μC se realizó usando el *Device initialization* de *CodeWarrior*. En la Figura 2-5 se pueden observar las características de la familia de microcontroladores MCF51QE, utilizando para este proyecto los dos canales de comunicación SCI (serial communication interface) y dos canales de ADC (analog to digital converter). La frecuencia de reloj de bus usada fue de 25 MHz.

- **Módulo ADC**

El módulo ADC es utilizado para la conversión analógica digital de las señales ECG y Pletismográfica. Se busca principalmente tener una buena frecuencia de muestreo y que ésta sea sincrónica, para evitar errores en las mediciones de tiempo de tránsito de pulso,

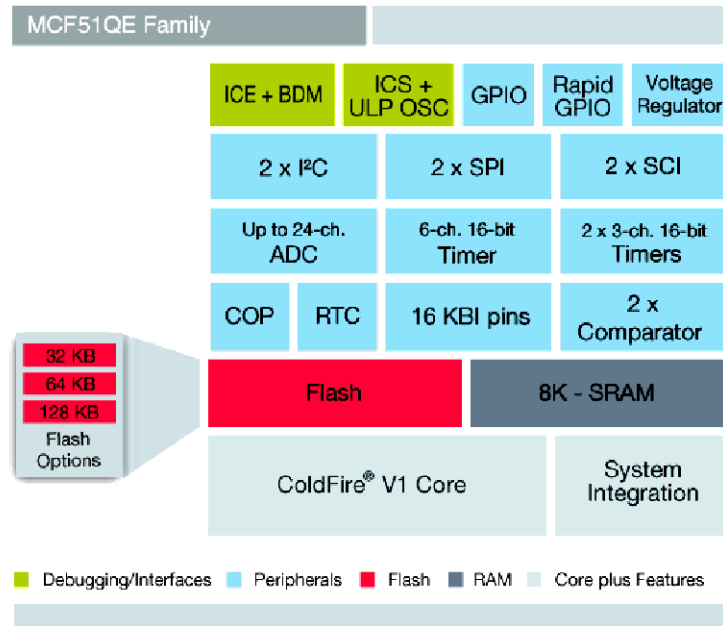


Figura 2-5: Familia de microcontroladores MCF51QE Freescale

en donde se toma la diferencia entre el pico de la onda QRS del ECG hasta el punto de ascenso de la onda PPG.

Para esto se programa la función *ReadADC*, la cual realiza una sola lectura de un canal del ADC cada vez que se ejecuta (el proceso de una lectura dura 6,56 us); el canal es seleccionado colocando en el argumento su número correspondiente. Si se ejecuta *ReadADC(14)* se está leyendo una muestra del valor actual del Electrocardiograma; en cambio, si se ejecuta *ReadADC(11)* se leerá una muestra de la onda Pletismográfica.

Los detalles de la programación del ADC en el microcontrolador se encuentran en el Apéndice 4.1.

■ Tasa de muestreo

Ya que cada paquete enviado contiene una muestra de las señales ECG, PPG y SpO_2 . La tasa de muestreo teórica para estas señales es igual a la relación entre la tasa de baudios³ al que se hace la comunicación y el número de bits que hay entre muestras.

³Representa el número máximo de cambios de estado que la señal puede tener en un segundo

Esta relación se muestra en la ecuación (2-1).

$$\begin{aligned} \text{Tasa de muestreo} &= \frac{19200 \text{ bits/s}}{40 \text{ bits}} = 480s^{-1} = 480Hz \\ \text{Periodo de muestreo} &= \frac{1}{480} = 2,08ms \end{aligned} \quad (2-1)$$

Para determinar esta frecuencia de forma empírica se dispone de un LED que cambia de estado por cada muestra enviada. Se mide esta frecuencia con un osciloscopio entre el terminal positivo del LED y tierra dando como resultado la imagen mostrada en la Figura 2-6, donde se puede observar que la frecuencia de señal es igual a 240,4 Hz. Si tenemos en cuenta que el envío de cada paquete produce un cambio de estado en el LED y que durante un periodo del LED hay dos estados '0' y '1'. Entonces la tasa de muestreo será igual al doble de la frecuencia del LED:

Tasa de muestreo = $2 * 240,4Hz = 480,8Hz$. Se comprueba así que la tasa de muestreo es por muy poco, la misma que se calculó.

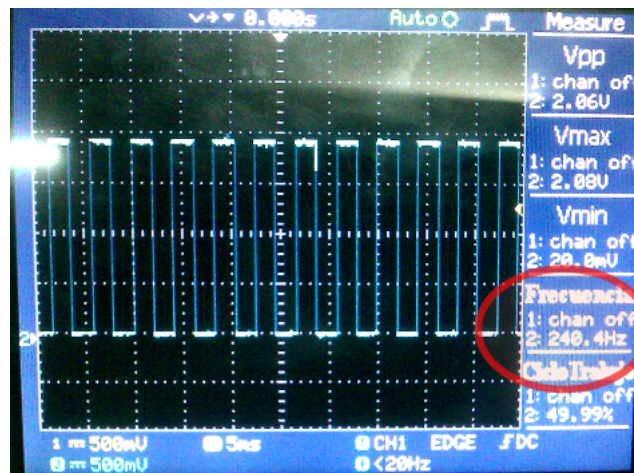


Figura 2-6: Medición de la frecuencia de señal en el LED

▪ Modulo SCI

Se usaron los dos canales de comunicación SCI del microcontrolador, el primero *SCI1* es usado para recibir los datos provenientes del pulsioxímetro, el cual tiene una tasa de

baudios de 4800; el segundo *SCI2* es usado para enviar los datos al PC con una tasa de baudios de 19200.

De forma general mientras se reciben datos (bytes) por un canal se envían por el otro, pero debido a la diferencia de tasas de baudios se debe evitar que el envío (que es mas rápido) tenga que esperar la recepción de un byte del pulsioxímetro (que es mas lento) ocasionando una transmisión de datos defectuosa. Para tal fin se debe ordenar al microcontrolador que siga enviando datos aún durante la espera que supone la recepción, es decir que mientras la bandera del registro de recepción serial no este lista se siguen enviando datos.

Otro paso a analizar es la extracción del byte de interés de la trama serial del microcontrolador, que en este caso es el concerniente al valor de saturación de oxígeno. Para tal fin se leen dos cadenas de datos del pulsioxímetro (10 bytes) y se hace la búsqueda del byte que contiene el bit de sincronización = 1 (*ver tabla 2-1*). Una vez hallado, se encuentra el byte de SpO_2 deseado avanzando cuatro cuatro posiciones. (*Apéndice 4.1*)

Finalmente se dispone el orden de envío de los datos al computador en forma serial sin paridad, empezando por una *bandera* o carácter de referencia usado para detectar cada inicio de una cadena de datos, que en este caso se ha escogido el numero 248, valor seleccionado por tener una baja frecuencia de aparición en la cadena de datos (ver Figura 2-7), evitando la detección errónea de otro byte que no sea la bandera.

Seguido a ésta bandera se envían los bytes correspondientes a *ECG*, *PPG* y *SPO₂*. (*figura 2-8*).

2.1.4. Diseño de la tarjeta principal

En el desarrollo del circuito impreso se tuvieron en cuenta las siguientes consideraciones:

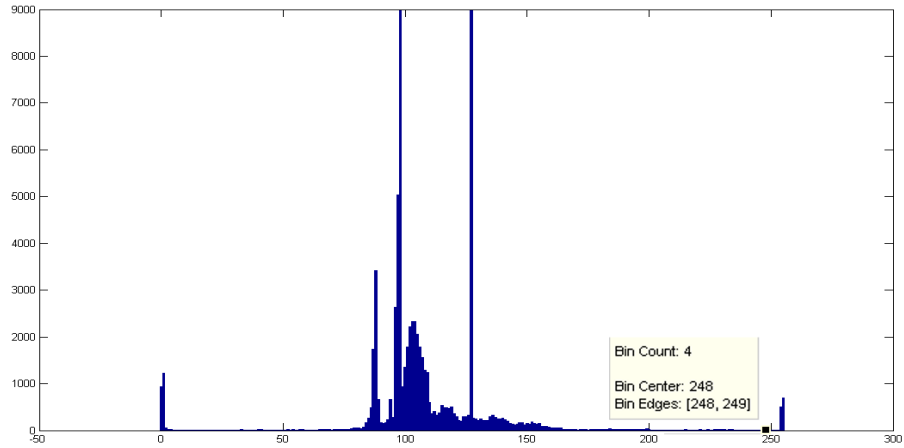


Figura 2-7: Frecuencia de aparición de valores de byte en 100000 bytes recolectados

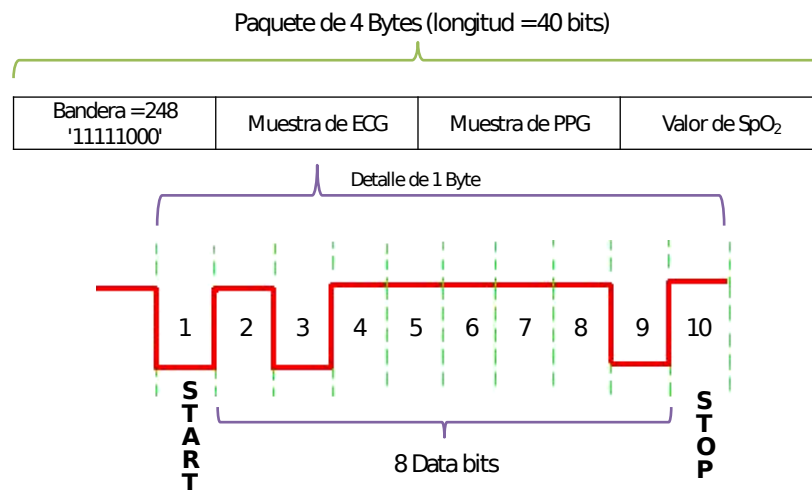


Figura 2-8: Representación del envío de un paquete por el microcontrolador hacia el PC

- La fuente de alimentación de el ECG, el microcontrolador y la comunicación esta dada por una fuente regulada de $\pm 3,3V$ derivada de los 5V DC que ofrece la alimentación USB y que a su vez energiza al pulsioxímetro. Es opcional la alimentación por batería de 9V pero para el desarrollo del proyecto se mantuvo solo por medio de USB.
- La adquisición analógica de la onda de pulso se hace directamente sobre el módulo de pulsioximetría a través de un *buffer* de alta impedancia de entrada para evitar perdida de señal en el módulo y luego ser llevada hacia el microcontrolador.

- La conversión de voltajes TTL a Serial y viceversa se hace posible mediante un IC MAX3232 que ofrece dos canales de conversión, uno para la recepción del pulsioxímetro y otro para el envío de datos al PC (*figura 2-9*).

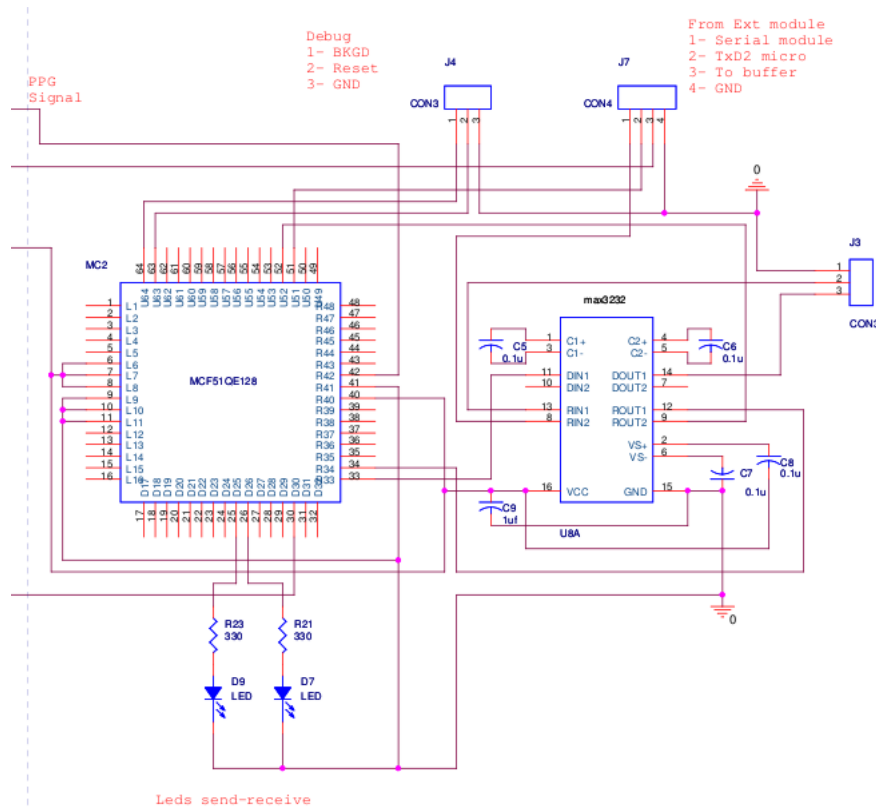


Figura 2-9: Esquemático del procesado y comunicación

- Los conectores se ubican lo mas cerca posible a los circuitos integrados, especialmente los correspondiente a las tomas del ECG, y la señal de pulso.
- Se dispone de planos de tierra en ambas caras y también una puesta a tierra del mallado del cable ECG para mejorar la respuesta al ruido.
- Se han dispuesto tres LEDS los cuales indican si esta energizado (LED rojo), si se están enviando datos (LED verde) y si esta el dedo no está correctamente puesto en el sensor de pulso (LED amarillo).

El *layout* del circuito impreso fue diseñado con el software *ORCAD Layout* y es mostrado en el Apéndice 4.1.

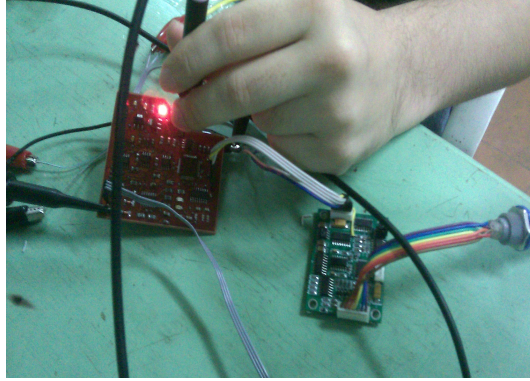


Figura 2-10: Pruebas realizadas a la tarjeta en el laboratorio

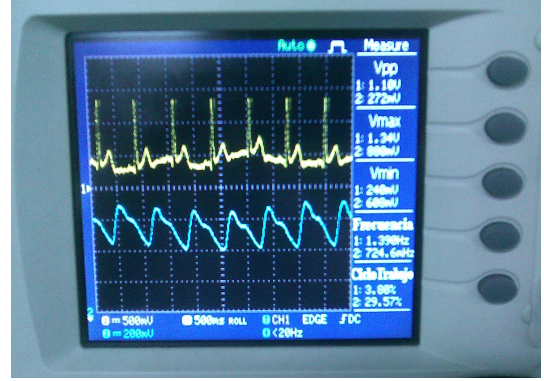


Figura 2-11: Visualización de las ondas PPG y ECG en el osciloscopio

2.1.5. Presentación final

El acabado final del prototipo además de ofrecer una mejora estética, cumple con la función de proteger los componentes internos y brindar un fácil traslado (*figura 2-13*). Para la medición de presión arterial se adquiere un esfigmomanómetro anerode marca *HOSUN*, mas un estetoscopio, además se adquieren 3 cables DIN 1,5 mm para ECG tipo *holter*. (*figura 2-12*)



Figura 2-12: Esfigmomanómetro y cable ECG tipo holter

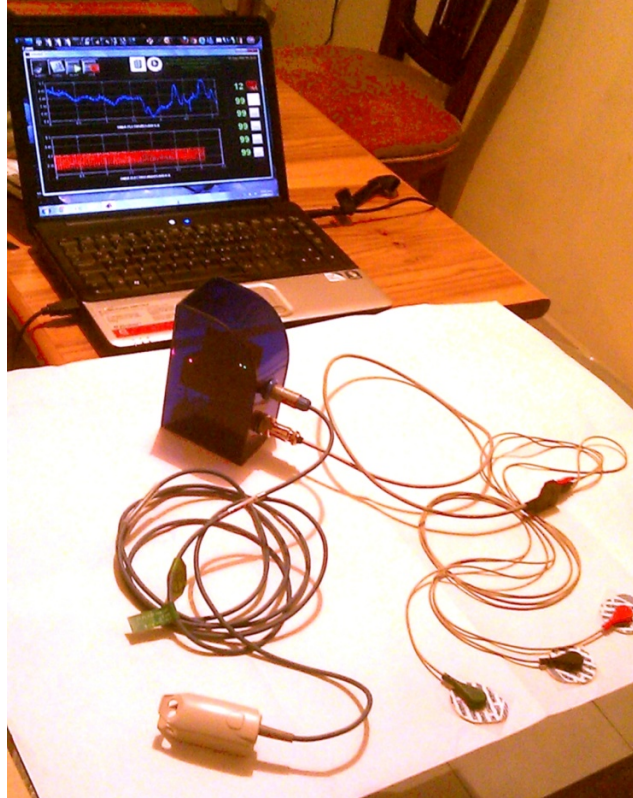


Figura 2-13: Presentación final del equipo de medición

2.2. Software

El software utilizado para la recepción de los datos, procesamiento de las señales y visualización de los resultados en una interfaz gráfica fue Matlab® con licencia de uso de la Universidad Industrial de Santander.

Los datos son enviados por el microcontrolador MCF51QE128 con formato RS232, y se reciben en el PC por medio de entrada USB. Matlab recoge y procesa dicha información. Cada paquete recogido contiene una bandera 248 y 3 muestras de datos: ECG, PPG y SpO_2 (véase Figura 2-8). Estas muestras son separadas y colocadas en vectores independientes. Por defecto Matlab recoge 1501 paquetes de datos (6004 bytes) cada vez que hace un llamado al puerto serial (por defecto COM2). Estos valores puede ser modificados en el código de Matlab (véase el apéndice 4.1).

En total son tres los vectores que contienen la información sobre onda de pulso, onda ECG y saturación de oxígeno respectivamente y con estos vectores se hace el tratamiento digital de señales: Filtrado de las señales, detección de picos, ondas dicroticas y puntos de ascenso, calculo de valores de frecuencia cardiaca (HR), tiempos de transito de onda de pulso (PWTT), volúmenes sistólicos (SV) y gastos cardiacos (CO). También se adecuan estos valores para evitar errores y se muestran en una interfaz gráfica hecha con la herramienta GUIDE de Matlab.

2.2.1. Filtrado

Para el filtrado de la señal de ECG se utilizaron dos tipos de filtro: Butterworth y de ventana en el dominio de la frecuencia, con el fin de obtener señales libres de ruido y de forma definida para facilitar la detección de picos y puntos importantes de la señal.

El filtro tipo Butterworth fue seleccionado por ofrecer una respuesta máximamente plana hasta su frecuencia de corte, la cual se debe escoger de tal manera que se eliminen componentes no deseados de frecuencia sin afectar la señal. Se ha seleccionado una frecuencia de corte $f_{corte} = 53Hz$, eliminando así la componente de 60 Hz sin afectar las frecuencias necesarias para el proyecto, correspondientes a frecuencias del complejo QRS, que van de 0,5 a 40 Hz (*véase Figura 2-14*). Para aplicaciones de monitoreo y detección de picos (objetivo del electrocardiógrafo diseñado), es suficiente con la banda de frecuencia mencionada [4].

Para obtener el filtrado deseado para la señal de ECG, en este proyecto se utilizaron filtros analógicos (filtro RC pasa-bajas con $f_{corte} = 40Hz$ de orden 1 y RC pasa-altas $f_{corte} = 0,5Hz$) y digitales (Butterworth pasa-bajas con $f_{corte} = 53Hz$ de orden 13 y de ventana en el dominio de la frecuencia pasa-altas con $f_{corte} = 0,5Hz$).

El espectro de frecuencias de la señal de ECG antes y después del filtrado pasa-bajas se muestra en la *Figura 2-15*. Se puede observar una gran atenuación del ruido de 60Hz.

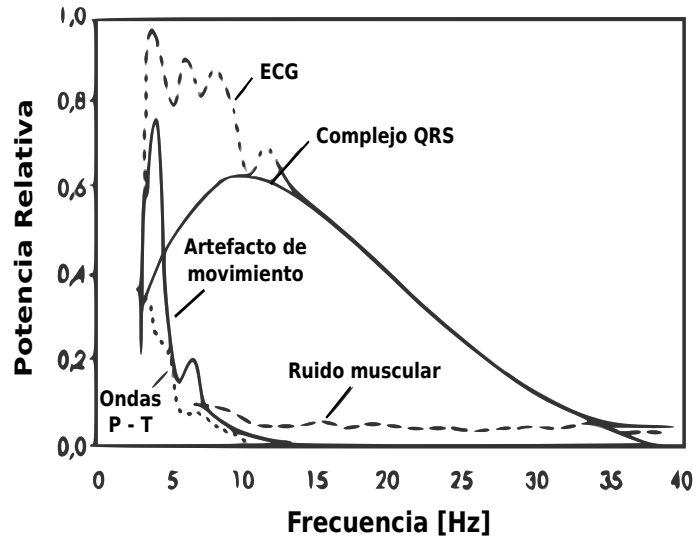


Figura 2-14: Espectro en frecuencia de la señal de ECG y el complejo QRS [4]

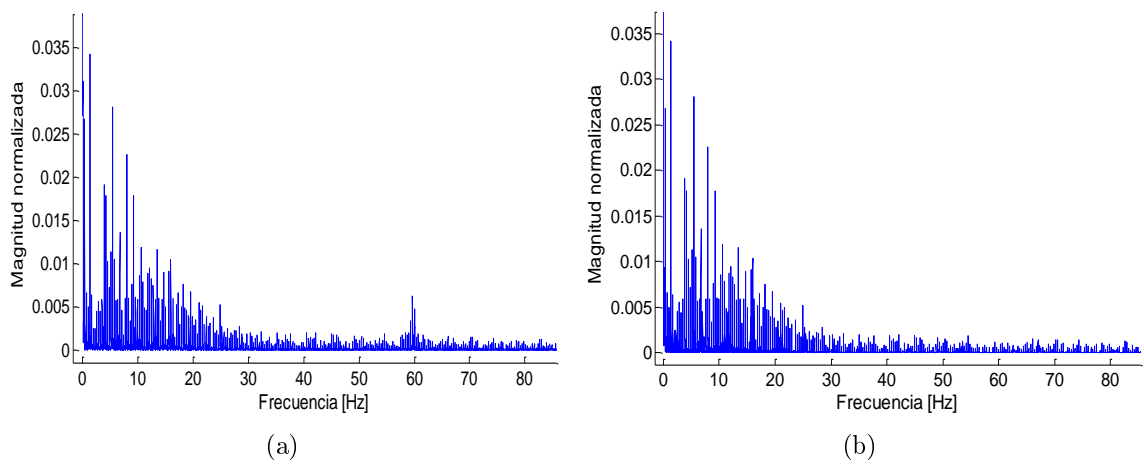


Figura 2-15: Filtrado pasa-bajas de la señal ECG. a) Sin filtrado b) Señal Filtrada

Para la efectiva detección de picos de la señal de ECG es necesario eliminar de ella las componentes de baja frecuencia (subidas y bajadas de nivel). Para tal fin se aplicaron los filtros pasa-altas mencionados, eliminando frecuencias por debajo de 0,5 Hz. Esto evitará cambios de nivel de la señal como se muestra en las Figuras 2-16a y 2-16b. Notese que en el espectro de frecuencias de la señal filtrada, observado en la Figura 2-16c, han sido atenuadas las frecuencias por debajo de 0,5 Hz.

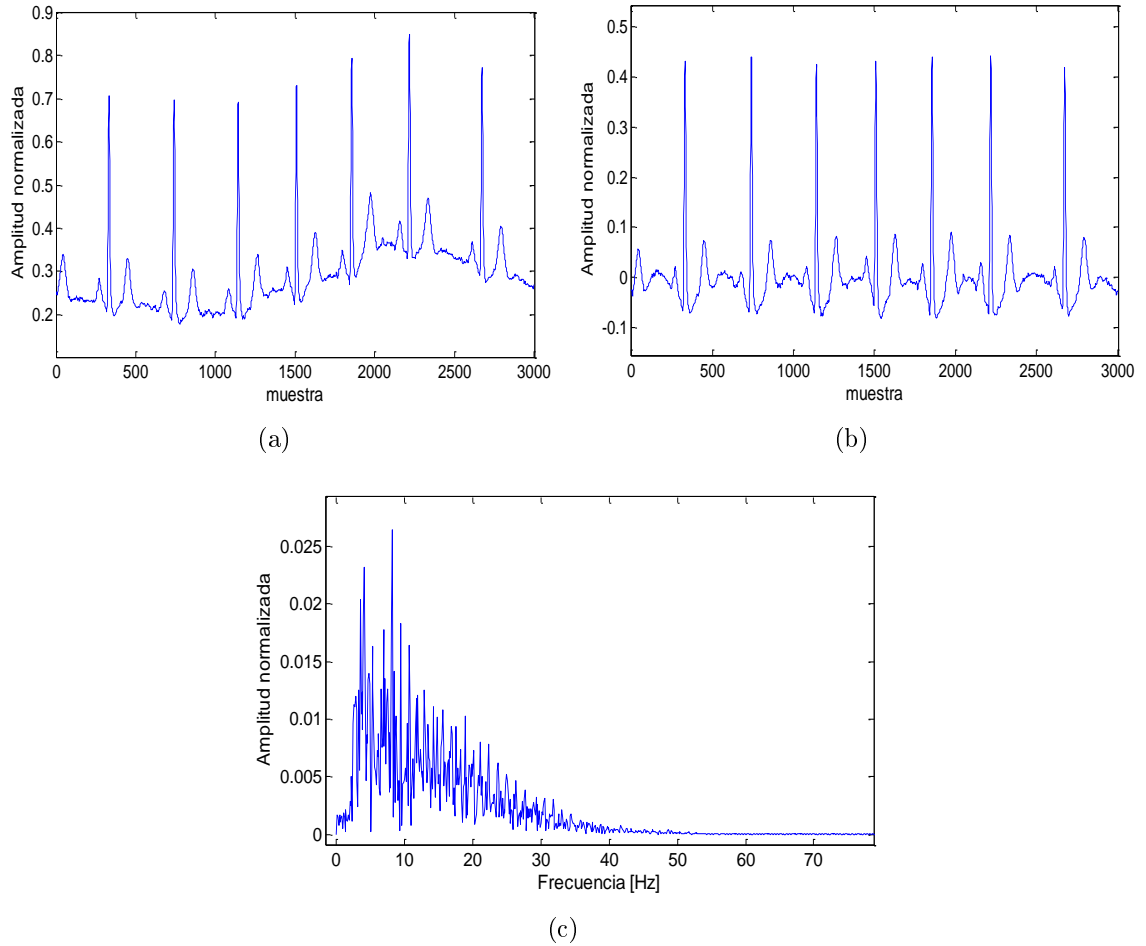


Figura 2-16: Filtrado pasa-altas de la señal ECG. En el tiempo a) Sin filtrado, b) Con filtrado y c) En frecuencia con filtrado

Aunque la señal PPG no presenta componentes de alta frecuencia (*figura 2-17a*), contiene picos muy pequeños y esporádicos (*figura 2-17b*) que no aportan suficiente energía para ser observados en el espectro de la señal. Utilizando un filtro Butterworth pasa-bajas de 53 Hz como se hizo con la señal de ECG estos picos desaparecen y además se evita la adhesión de señales indeseadas en ambientes contaminados con ruido de alta frecuencia o componente de 60 Hz. El resultado de la señal filtrada se muestra en la *Figura 2-17*.

Al igual que con la señal de ECG, para la señal pletismográfica PPG se aplicó

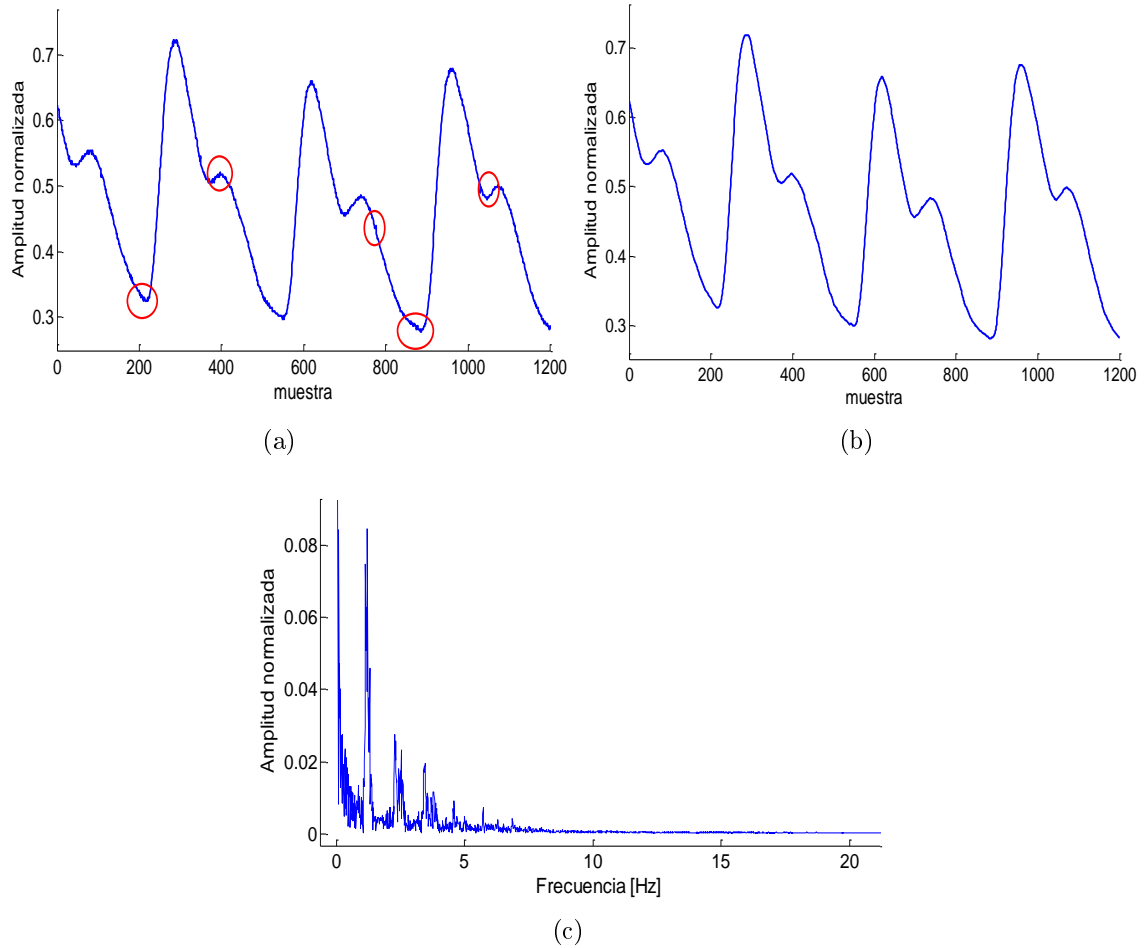


Figura 2-17: Filtrado pasa-bajas de la señal PPG. En el tiempo a) Sin filtrado (picos esporádicos resaltados), b) Con filtrado y c) En frecuencia con filtrado

un filtro digital pasa-altas de ventana en frecuencia con $f_{corte} = 0,1Hz$ para eliminar componentes de baja frecuencia que produzcan cambios de nivel en la señal y dificulten el proceso de detección de picos.

2.2.2. Detección de picos

El método utilizado para la encontrar máximos y mínimos de las señales es la detección por umbral dinámico, una vez que una sección de la señal supere el umbral definido, se aplican la funciones $max()$ o $min()$ de Matlab para determinar el máximo o mínimo local y se almacena el índice en el vector correspondiente. El procedimiento

para determinar los máximos es descrito por medio del diagrama de flujo mostrado en la Figura 2-18.

Los umbrales definidos para la detección de máximos y mínimos de ECG y PPG son respectivamente:

$$\text{Umbral de detección para máximos de ECG} = 1,15 * (\text{máximo} + \text{media}) / 2$$

$$\text{Umbral de detección de PPG, para máximos} = 1,10 * (\text{máximo} + \text{media}) / 2$$

$$\text{para mínimos} = 0,90 * (\text{mínimo} + \text{media}) / 2$$

Los umbrales se determinaron de manera experimental. Para la señal de ECG se buscó un punto entre el picos y el cuerpo de la onda (parte de la onda que no pertenece al complejo QRS), igualmente para la onda de PPG. Los umbrales sobre las gráficas de ECG y PPG son mostrados respectivamente en las Figuras 2-19 y 2-20.

Los resultados obtenidos en la detección de los picos se muestran en la Figura 2-21 para una señal ECG estable y en la Figura 2-22 para una señal de ECG con ruido electromiográfico y cambios de nivel en la onda. En las figuras se pueden observar los umbrales definidos por el algoritmo (línea horizontal verde).

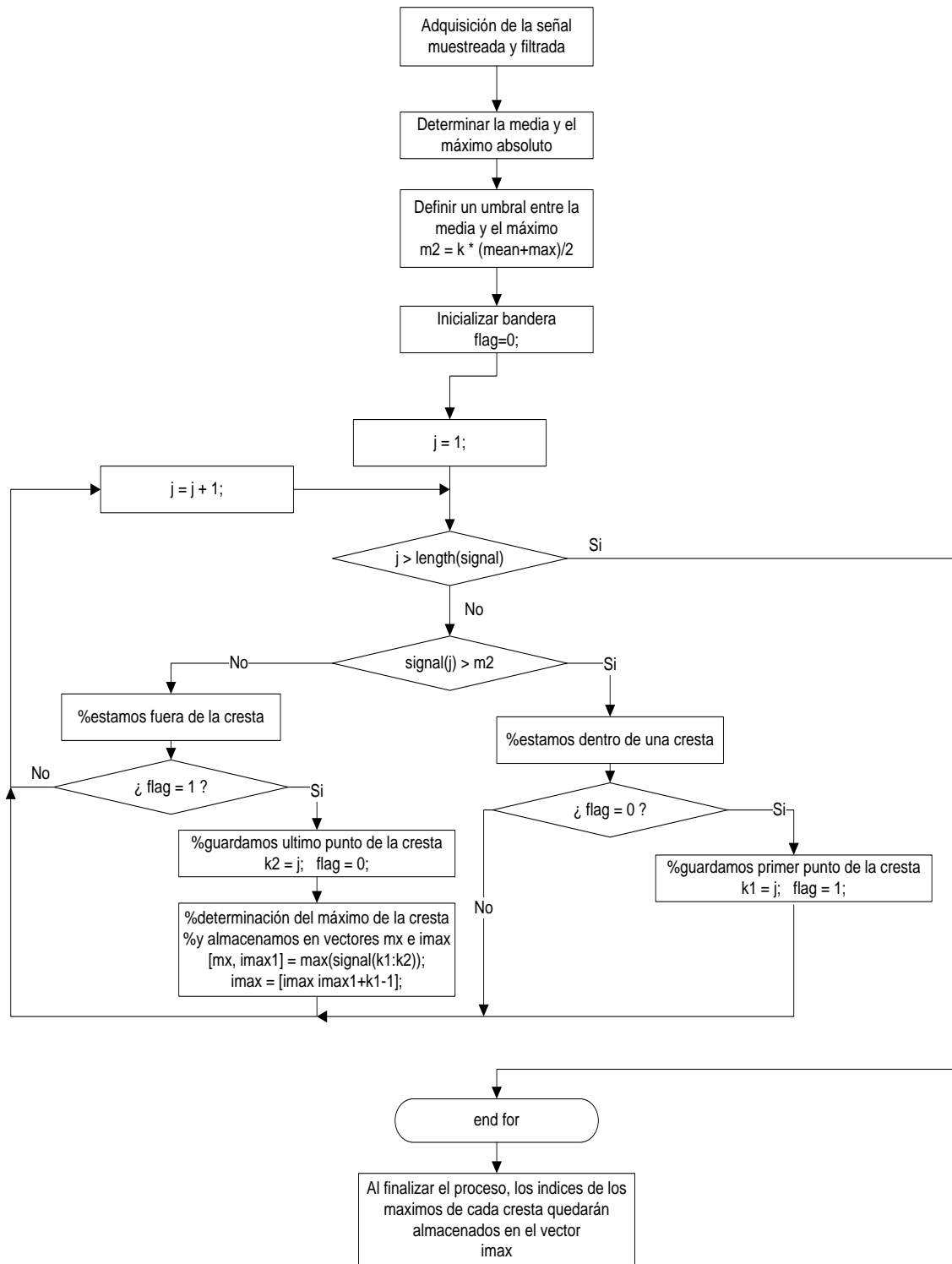


Figura 2-18: Algoritmo de detección de picos

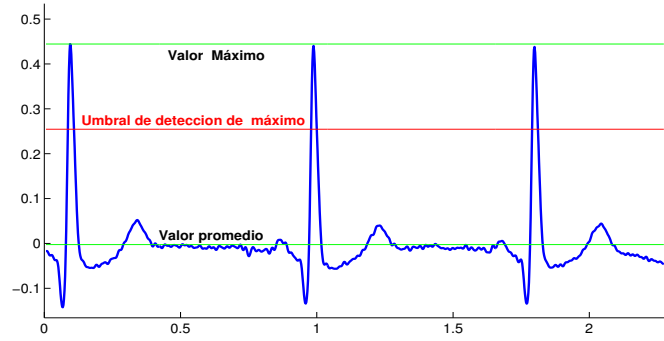


Figura 2-19: Visualización de umbral superior (color rojo) para detección de picos en la señal de ECG

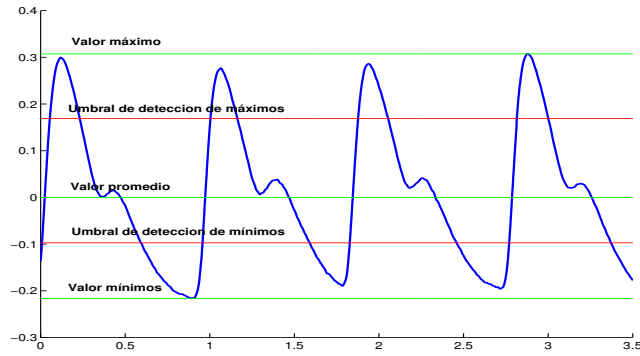


Figura 2-20: Visualización de umbrales superior e inferior (rojos) para detección de picos en la señal de PPG

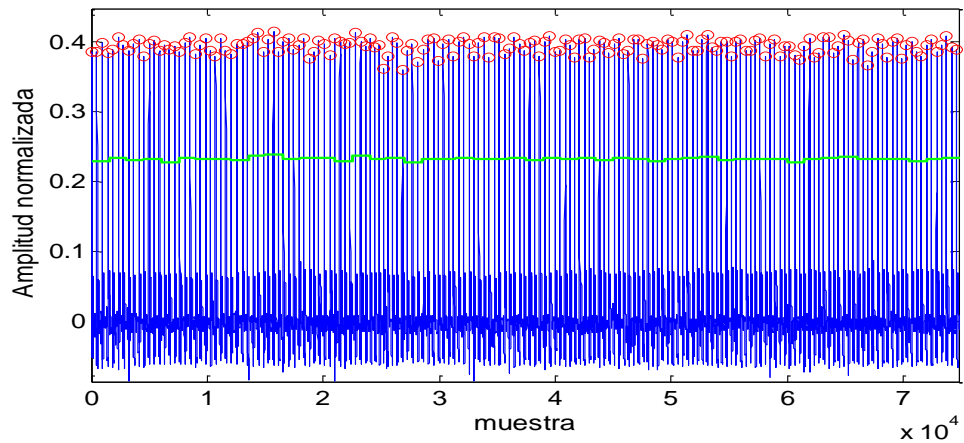


Figura 2-21: Muestra de picos detectados por algoritmo para una señal estable. Detectados 174 de 174 picos (error = 0 %) para una señal de 76500 muestras (duración: 160 segundos)

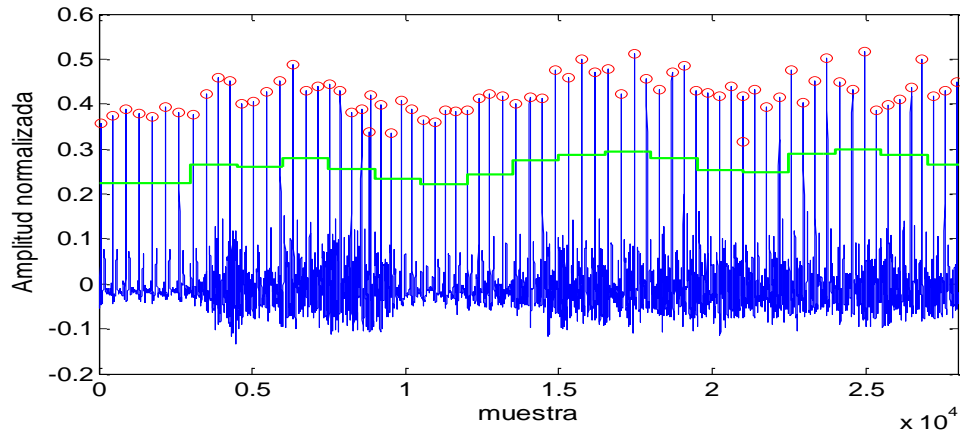


Figura 2-22: Muestra de picos detectados por algoritmo para una señal con ruido electromiográfico y cambios bruscos de nivel. Detectados 207 de 205 picos (0 ignorados, 2 duplicados; error de 0,98 %), para una señal de 79500 muestras (duración: 165 segundos)

2.2.3. Detección de punto de ascenso y onda dicrótica

Para la detección del punto de ascenso de la onda pletismográfica PPG, se debe encontrar el momento en que la derivada alcanza un 30 % de su valor máximo, conceptualmente se muestra en la sección 3.1. El procedimiento que realiza el algoritmo es, una vez encontrado el mínimo, seguir el valor de la primera derivada hasta que sea mayor o igual al 0,3 veces del máximo de la misma, en ese momento se guarda el índice del punto correspondiente en un vector (*véase el código detallado en el Apéndice 4.1, i3dppg.m*).

El procedimiento para encontrar el punto dicrótico de la onda es: Conocidos los máximos y los mínimos de la onda PPG es necesario organizarlos por pares consecutivos, cada vez que aparezca un par máximo-mínimo se asume que hay una onda dicrótica entre ellos. Se hace entonces un barrido de izquierda a derecha (de máximo a mínimo) y el primer punto que tenga derivada mayor o igual a cero será la dicrótica, si por este método no se encuentra dicrótica, el segundo criterio es calcular la máxima derivada de la porción de onda (de máximo a mínimo) y el primer punto con derivada mayor o igual a 0,35 veces la máxima derivada será la dicrótica. La detección de los puntos dicróticos en una onda PPG común se visualizan en la Figura 2-23, el código detallado

para la detección de la onda dicrótica se encuentra en el *Apéndice 4.1, idicppg.m*.

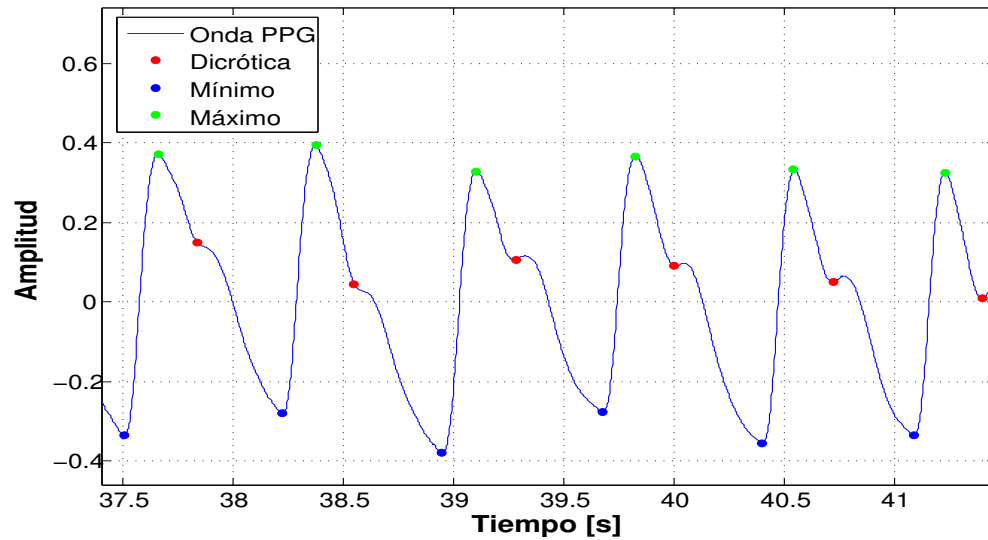


Figura 2-23: Puntos dicróticos detectados en una onda PPG típica

2.2.4. Frecuencia cardíaca

Para el cálculo de la frecuencia cardíaca (HR) se comprueba que el número de picos obtenidos de la señal de ECG se encuentre entre 2 y 11 (frecuencias cardíacas válidas entre 20 y 211 latidos por minuto), si se cumple esta condición se procede a calcular la frecuencia cardíaca como la 60 por el periodo de muestreo sobre la media de distancia entre picos (distancia en muestras), como se muestra en la ecuación 2-2

$$HR = \frac{60 * mps}{\text{mean}(\text{distancia entre picos})} \quad (2-2)$$

En caso de no cumplirse las condiciones para la señal de ECG, se procede a hallar el ritmo cardíaco con la señal PPG, comprobando que sea válida (valor de SpO_2 sea diferente de 127) y que el número de máximos y mínimos sea igual a 4 o más y se calcula de la misma manera que se hizo para la señal de ECG. El código detallado es mostrado en el *Apéndice 4.1, Cálculo de frecuencia cardíaca*.

2.2.5. Cálculo del PWTT

Para calcular el tiempo de tránsito de la onda de pulso PWTT se verifica que haya un mínimo de tres máximos de la onda de ECG y tres puntos de subida de la onda de PPG detectados para realizar la comparación. El PWTT se calcula como el tiempo entre el pico R del complejo QRS de la onda de ECG y el punto de subida de la onda pletismográfica PPG.

Se comprueba cual de los dos picos aparece primero, si aparece primero un máximo de ECG se calcula el tiempo tomando la diferencia entre el segundo punto de subida de PPG y el segundo máximo de ECG. En el caso contrario, de aparecer primero un punto de subida de PPG, se calculará tomando la diferencia entre el tercer punto de subida de PPG y el segundo máximo de ECG. El resultado del calculo es mostrado en la Figura 2-24 y el código se encuentra en el *Apéndice 4.1, Cálculo del PWTT*)

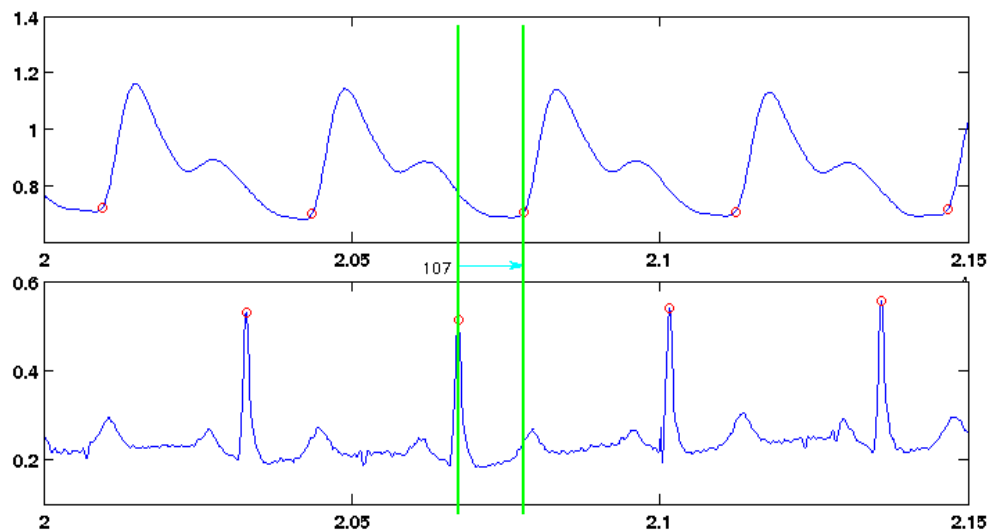


Figura 2-24: Determinación del tiempo de tránsito de pulso entre las señales de ECG y PPG.

2.2.6. Validación

Para validar el dispositivo del proyecto se compararon las señales adquiridas con el equipo PowerLab 16/30 (mostrado en la *figura 2-25*) perteneciente a la facultad de

salud de la UIS y las adquiridas con el dispositivo a implementar, tomadas ambas en paralelo sobre uno de los autores del proyecto.



Figura 2-25: Equipo de validación Powerlab 16/30

El Powerlab 16/30 es un sistema de adquisición de datos de alto desempeño, con capacidad de manejar hasta 16 canales y grabar a velocidades de 400.000 muestras por segundo de forma continua en el disco. Para la prueba se configuró a una tasa de muestreo de 1000 muestras por segundo, con filtrado pasa-altas de 0,5 Hz y pasa-bajas de 45 Hz para asemejar las condiciones de prueba. Una muestra de las señales adquiridas con ambos equipos se muestra en la Figura 2-26.

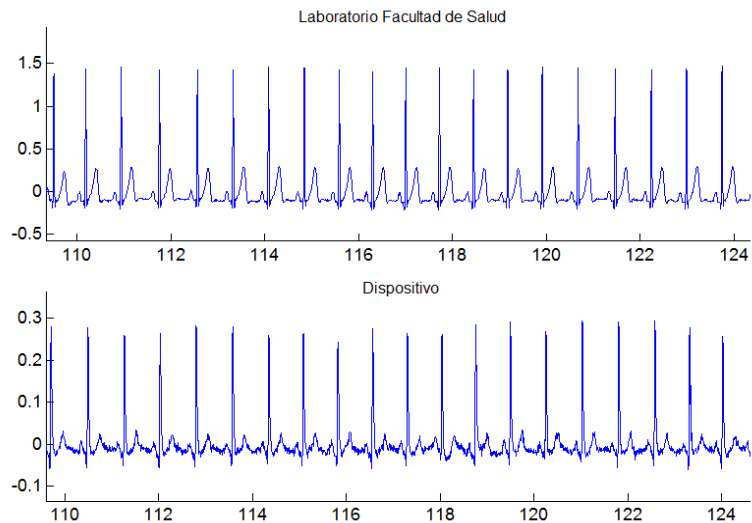


Figura 2-26: Fragmento de señales adquiridas durante la validación

Los tiempos entre picos R-R de la onda electrocardiográfica no son constantes, es normal que a medida que avanza el tiempo muestre una forma ondulatoria debido al ciclo respiratorio (proceso conocido como arritmia fisiológica respiratoria), durante la inspiración el tiempo entre intervalos R-R se acorta y durante la inspiración se alarga, teniendo variaciones comunes entre el 30 y el 40 %. [14]

Debido a este fenómeno y a las condiciones de hardware, muestreo y comunicación, el tiempo entre picos de la señal adquirida con el equipo PowerLab 16/30 cubre un rango de valores entre 0,616 y 0,857 segundos, con una media igual a 0,7365 y exhibiendo una desviación estándar igual a 0,0447 segundos. Mientras que el tiempo entre picos para la señal adquirida con el dispositivo tiene un rango de valores entre 0,628 y 0,871 con media igual a 0,7591 y desviación estándar de 0,0465 (*figura 2-27*).

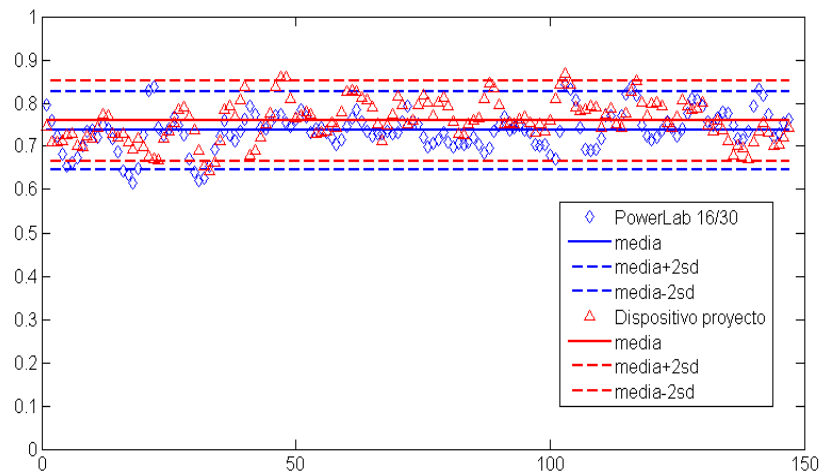


Figura 2-27: Tiempo entre picos (en segundos) para las señales adquiridas con el equipo PowerLab 16/30 (diamantes azules) y el dispositivo de proyecto (círculos rojos) para una muestra de 111,7 segundos (147 picos) de ambas señales

PowerLab 16/30 (azul)

Valor medio = 0,7365

Desviación estándar = 0,0447

Dispositivo proyecto (rojo)

Valor medio = 0,7591

Desviación estándar = 0,0465

Para el cálculo del ritmo cardíaco uso la ecuación 2-2 tomando el promedio de la distancia cada 6 picos de ECG. Dando como resultado 24 valores de ritmo cardíaco

para cada señal. El valor medio del ritmo cardiaco determinado a partir de la señal azul (adquirida del equipo PowerLab 16/30) es igual a 79,29 con una desviación estándar de 3,57. El valor medio calculado a partir de la señal roja (dispositivo a validar) tiene media de 81,82 y desviación igual a 3,144, dando como resultado un porcentaje de error entre medias obtenido igual a 3,09%. Estos resultados son mostrados en la *figura 2-28*

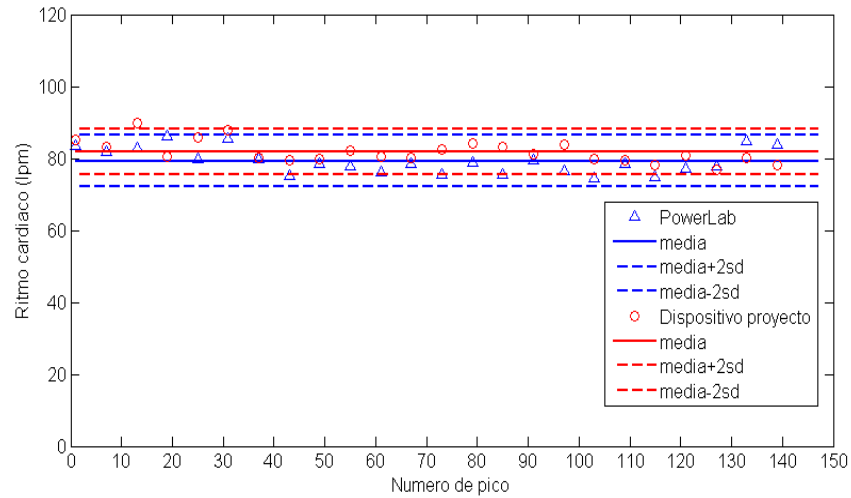


Figura 2-28: Comparación de ritmo cardiaco de acuerdo a los tiempos entre picos mostrados en la Figura 2-27, para cada equipo

Comparación en el cálculo del PWTT

Para comparar los resultados obtenidos entre tiempos de transito de onda de pulso para ambos equipos (un fragmento de las gráficas son mostradas en la *figura 2-29*) se usa el gráfico de Bland Altman, en el cual los puntos tienen un acuerdo promedio perfecto si tienen diferencia cero (en $y=0$). Se obtuvo con el método de Bland Altman una media de las diferencias igual a $-0,01659$, como se muestra en la Figura 2-30.

Para el cálculo de los límites de acuerdo del 95 % al promedio se le resta y suma 1,96 veces la desviación estándar. Se obtienen así intervalos de acuerdo del 95 % entre 0,0322 y $-0,0010$ para las diferencias entre los dos equipos. El error estándar para los limites de acuerdo es igual a $8,4078 * 10^{-5}$, valor que fue calculado con la ecuación 2-3

$$Er_{95\%} = \frac{\sqrt{3 * sd^2}}{n} \tag{2-3}$$

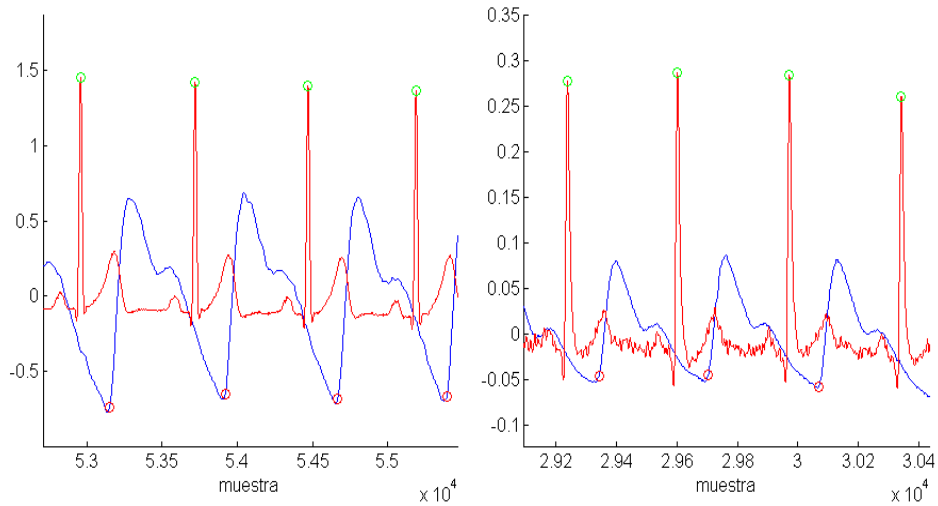


Figura 2-29: *Izquierda. Ondas ECG y PPG de equipo PowerLab 16/30. *Derecha. Comparación ondas ECG y PPG del dispositivo

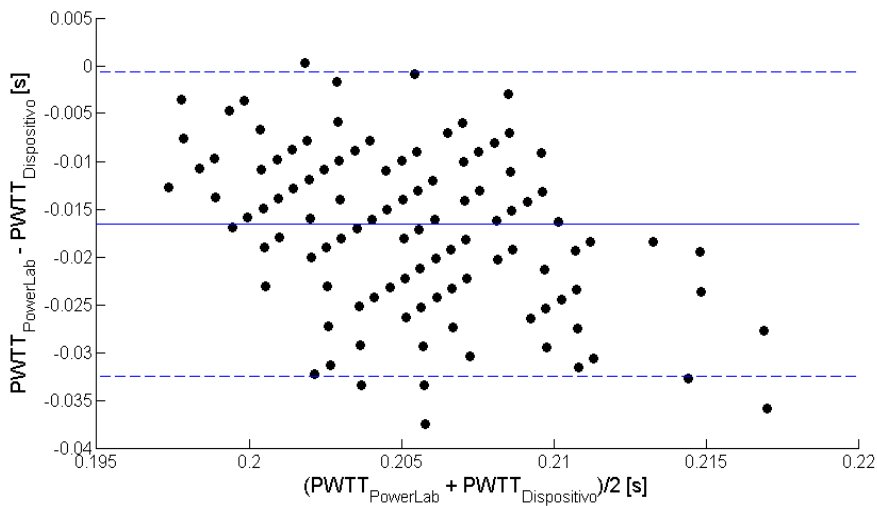


Figura 2-30: Gráfico de Bland Altman para comparación de tiempos de transito de pulso entre los dos equipos

2.2.7. Interfaz gráfica con Matlab

La visualización de los datos esta hecha con la herramienta GUIDE de Matlab, en donde se pueden apreciar en la parte superior 5 botones establecidos asi:

- PORT: permite seleccionar el puerto de comunicación en el computador, con

opción de 10 puertos COM.

- CONFIG: pulsando este botón aparece un formulario de ingreso de datos del usuario como: nombre, edad, peso, estatura y presión arterial.
- START: inicia la recepción de datos y activa la visualización de las ondas ECG y PPG.
- STOP: cancela la recepción de datos y se da por terminada la medición. Adicionalmente se activa la opción de medir tiempos entre cualquier lugar de las dos ondas presentes.
- REC: pulsando este botón se inicia la grabación de datos hasta que sea oprimido nuevamente.

En la esquina superior derecha se visualiza la hora y fecha de la medición; en la zona lateral derecha se observan las medidas de frecuencia cardíaca, saturación de oxígeno, PWTT, gasto cardíaco y volumen sistólico por diferentes métodos. En el área central se encuentran dos ejes coordenados para visualizar la onda ECG y PPG y en el área inferior derecha un recuadro para medir distancia entre puntos y amplitudes.

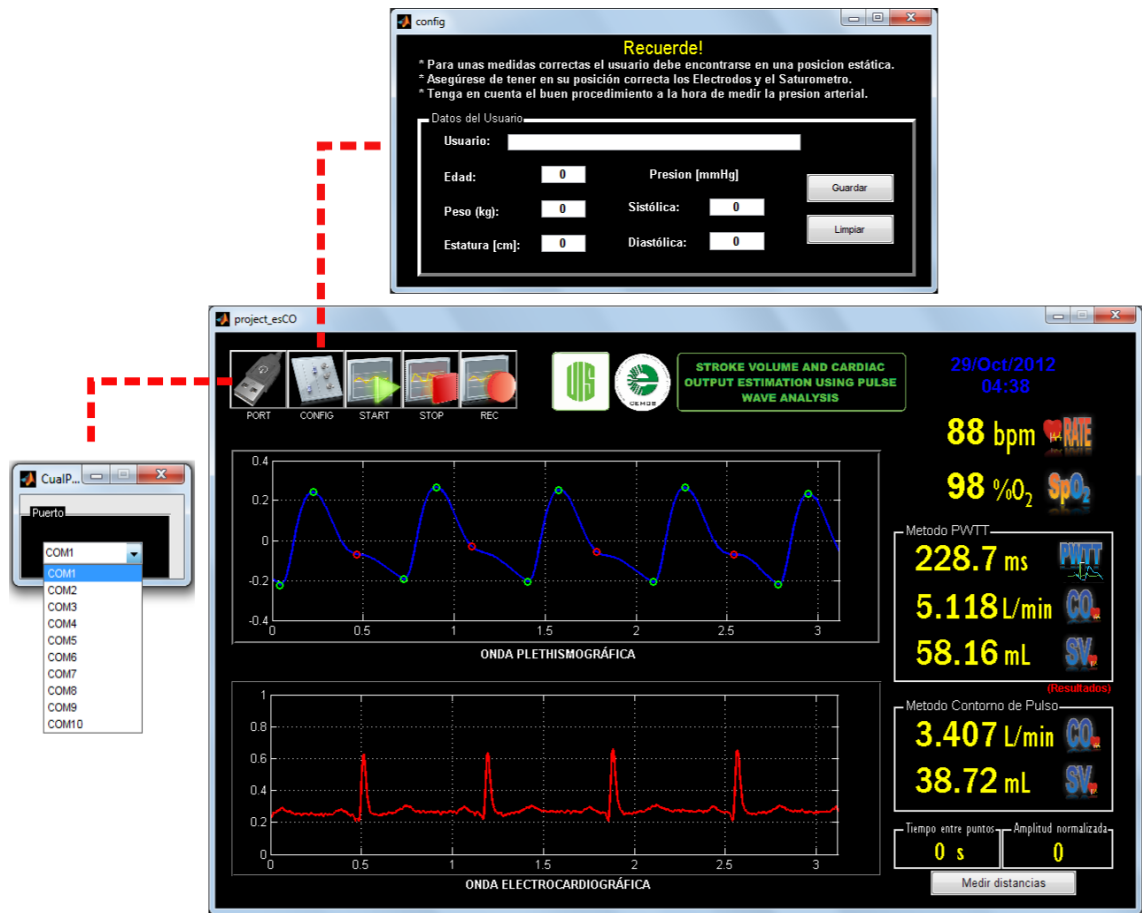


Figura 2-31: Interfaz gráfica del equipo de estimación de volumen sistólico y gasto cardiaco

Capítulo 3

Metodología y análisis de datos

Existen métodos considerados no invasivos como la ecocardiografía Doppler, reinhalación de Oxígeno, bioimpedancia transtorácica y análisis de la onda de presión arterial, pero de forma más reciente un grupo de investigación Japonesa asociado Nihon Khoden (desarrolladores de la pulsioximetría) han utilizado el PWTT (tiempo de tránsito de pulso), encontrando en éste una relación significativa con los cambios de SV (volumen sistólico) del cual se deriva el gasto cardiaco. El análisis de onda de pulso también hace parte de investigaciones enfocadas a determinar volumen sistólico, debido a la relación existente con el flujo sanguíneo, la resistencia vascular, la elasticidad arterial y la presión arterial.

3.1. Tiempo de transito de la onda de pulso

El tiempo de tránsito de pulso se definió como el tiempo entre el pico de la onda R en el ECG al punto de ascenso de la onda de pulso. Tal punto se define en este proyecto como el 30 % de la amplitud pico de la primera derivada de onda de pulso, debido a que no se considera el mínimo de la onda PPG como el verdadero punto de llegada, esta afirmación se basa en la forma en que la onda de pulso es afectada por la velocidad del flujo sanguíneo, la propagación de pulso y la reflexión causada desde la periferia [15].

El PWTT está dividido en tres intervalos conocidos como el periodo de pre-eyección

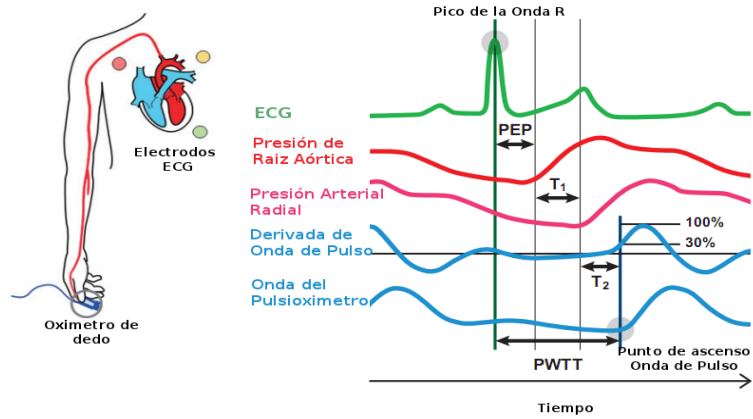


Figura 3-1: Componentes del tiempo de tránsito entre la onda ECG y la onda de pulso. [5]

(PEP), el tiempo de tránsito a través de la arteria elástica (T_1), y el tiempo de tránsito a través de las arterias periféricas (T_2). El PEP está definido como el tiempo desde el pico de la onda R hasta el inicio de la onda de presión en la raíz aórtica. T_1 fue definido como el tiempo de inicio de la presión en la raíz aórtica hasta el inicio de la presión arterial radial y T_2 como el tiempo de inicio de la presión arterial radial hasta el punto de ascenso en la onda PPG en el dedo. (figura 3-1) [16].

$$PWTT = PEP + T_1 + T_2 \quad (3-1)$$

■ **Relación entre el PEP y SV**

El estudio de Weissler demostró una correlación negativa entre el PEP y el SV. Donde:

$$SV = c_0 \cdot PEP + d_0$$

Donde c_0 y d_0 son constantes inherentes.

■ **Relación entre el T_1 y SV**

Se determina que el volumen sistólico es inversamente proporcional a la raíz de T_1 y un incremento en el volumen sistólico representa un incremento de la presión sanguínea acompañado de una reducción del tiempo T_1 .

El volumen sistólico se expresa como:

$$SV = \frac{CO}{HR} = \frac{1}{HR} \cdot \frac{P}{R_v} \quad (3-2)$$

donde $P = V_1^2 \cdot \frac{\rho b}{Q_0} \cdot Q = V_1^2 \cdot c$

$$V = T_1^{-1}$$

$$SV = \frac{c}{R_v} \cdot \frac{1}{HR} \cdot \frac{1}{T_1^2}$$

R_v = resistencia vascular periférica.

■ **Relación entre el T_2 y SV**

De la ecuación de Poiseuille R_v es expresado como:

$$R_v = \frac{1^4}{r} \cdot c_2 \quad (3-3)$$

donde $c_2 = 8\mu L/\pi$

$$(3-4)$$

μ : viscosidad y L: longitud del vaso sanguíneo.

El diámetro del vaso sanguíneo cambia de acuerdo a la diferencia de velocidades de onda de pulso arterial V_2 y pulso periférico [16]. Llegando a la siguiente ecuación:

$$\frac{1}{r} = \frac{1}{c_1} \cdot \frac{V_1 - V_2}{V_1} \quad (3-5)$$

donde $c_1 = \sqrt{\frac{v}{2\omega}}$

v = viscosidad cinemática; ω = frecuencia angular de la onda de presión arterial

combinando 3-3 y 3-5 se llega a:

$$Rv = c_3 \cdot \left(\frac{V_1 \cdot T_2 - 1}{V_1 \cdot T_2} \right)^4$$

donde $c_3 = c_2/c_1^4$

luego:

$$\begin{aligned} SV &= \frac{CO}{HR} = \frac{1}{HR} \cdot \frac{P}{R_v} = c_4 \cdot \frac{P}{HR} \cdot \left(\frac{V_1 \cdot T_2 - 1}{V_1 \cdot T_2} \right)^4 \\ &= c_4 \cdot \frac{P}{HR} \cdot \left(1 + \frac{V_1 \cdot T_2 - 1}{V_1 \cdot T_2} \right)^4 \end{aligned} \quad (3-6)$$

donde $c_4 = 1/c_3$

Por lo tanto el volumen sistólico es inversamente proporcional a T_2 . Un incremento en SV con un decremento en R_v conlleva un decremento de T_2 .

■ Aproximaciones

Debido a que los cambios de los tiempos T_1 y T_2 son pequeños comparados al $PWTT$, una aproximación lineal es considerada posible. Por lo tanto el volumen sistólico es expresado como:

$$SV = \alpha * (PEP + T_1 + T_2) + \beta = \alpha * PWTT + \beta \quad (3-7)$$

Para calcular el valor de α y β al menos dos pares de SV y PWTT son necesarios y deben ser recolectados cuando el SV cambie, pero no se considera muy práctico calibrar por cambios de volumen sistólico, así que se recurre a una constante K que relacione el SV y la presión arterial PP, donde $K=SV/PP$. Sustituyendo esto en la ecuación 3-7:

$$SV = K * PP = K * (\alpha' * PWTT + \beta') \quad (3-8)$$

donde alfa es fijado y K y β son determinados [16]. α es determinado de acuerdo a los cambios entre PP y PWTT encontrando una constante de $-0,30$ para humanos y el valor de β es calculado según la ecuación $PP = \alpha' * PWTT + \beta'$ [17], siendo una variable para cada persona.

El gasto cardiaco se deriva de la ecuación 3-7 como:

$$CO = K * (\alpha * PWTT + \beta) * HR \quad (3-9)$$

Donde el valor de K es derivado del PWTT, HR y CO al comienzo de la medición en cada paciente de acuerdo a la ecuación 3-9. Por lo tanto una medida de CO al inicio es requerida, utilizando ecocardiografía o en su defecto se calibra con datos del paciente (método utilizado en este proyecto) [17].

3.2. Análisis de onda de pulso

Desde los años 60 varios estudios sobre la estimación de gasto cardiaco han sido derivados del análisis de onda de pulso. Si se conocen la onda de presión y la impedancia característica del paciente es posible determinar el flujo sanguíneo por pulso, y al integrar esto en el tiempo se puede lograr un estimado del volumen sistólico [7]:

$$SV = \int_0^T q(t)dt = \int_0^T \frac{p(t)}{Z_c} dt = \frac{1}{Z_c} \int_0^T p(t)dt \quad (3-10)$$

El gasto cardiaco se deriva como: $CO = SV * HR$

Una forma de obtener una onda de presión es calibrando la onda de pulso teniendo en cuenta los valores de presión arterial diastólica y sistólica. Dada la semejanza entre la onda pletismográfica y la onda de presión con un Δ -amplitud de onda casi constante

en el tiempo, se hace posible esta calibración y se analiza su forma de onda, originando una constante K relacionada con el área bajo la curva de la onda de pulso. La similitud entre el comportamiento de la onda pletoisográfica y la onda de presión arterial se puede apreciar en la Figura 3-2 [6].

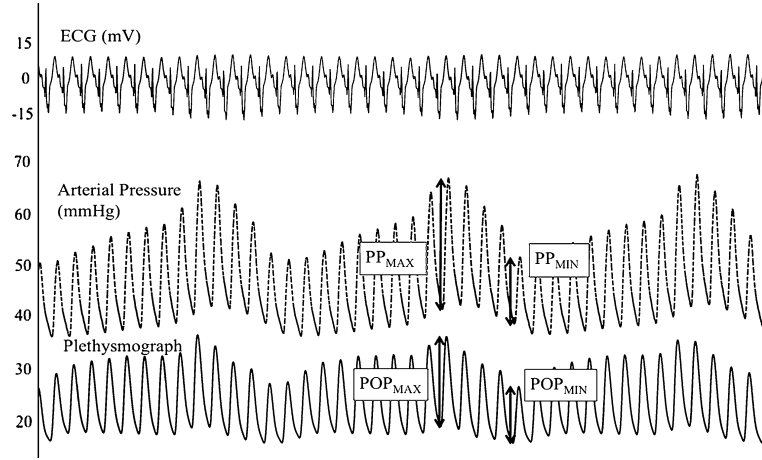


Figura 3-2: Comportamiento de la onda de presión arterial frente a la onda pletoisográfica [6].

$$K = \frac{P_m - P_d}{P_s - P_d} \quad (3-11)$$

donde P_m es la presión media.

Se asume para un humano en condiciones fisiológicas normales una constante K de 0,35 [7]. Cuando la resistencia periférica, la viscosidad en la sangre o la elasticidad vascular cambia, el valor de K tiene su correspondiente cambio, para este proyecto éste valor varía de 0,35 a 0,6 dependiendo de la forma de onda y sus presiones sistólica y diastólica, formulando el volumen sistólico como:

$$SV = \frac{17}{K^2 * HR} * (P_s - P_d) \quad (3-12)$$

$$CO = SV * HR = \frac{17}{K^2} * (P_s - P_d) \quad (3-13)$$

En donde los valores de K , HR , P_s y P_d son fácilmente asequibles por medios no invasivos.

Pero es posible tener el mismo valor K para diferentes formas de onda, lo que convierte a este método en equivoco en algunos casos. Se puede corroborar tal situación, evaluando la ecuación 3-11 en las formas de onda de la Figura 3-3.

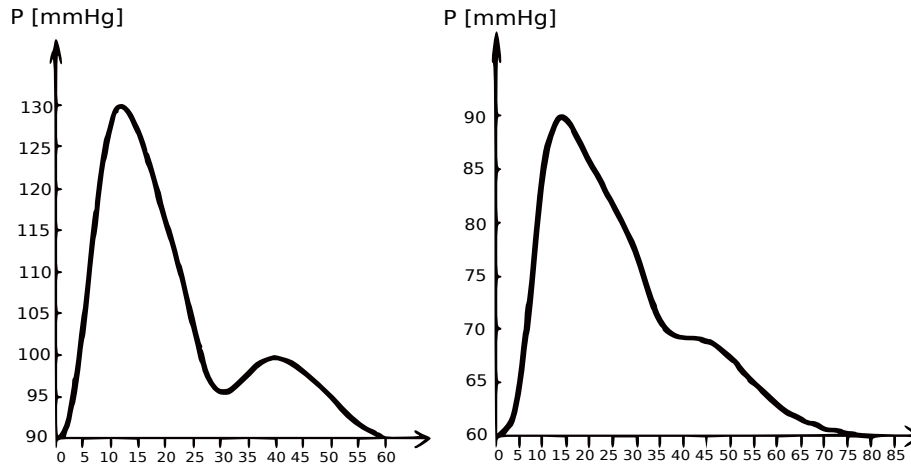


Figura 3-3: Diferentes formas de onda con el mismo valor de constante $K = 0,35$. [7]

La forma de mejorar el método es dividiendo el área bajo la curva en dos, obteniendo así dos valores de K diferentes K_1 y K_2 (figura 3-4). K_1 es derivado de la razón entre el área OAB y el área OCDG, mientras K_2 se deriva de la razón entre el área BF y el área DEFG. Así que ahora se pueden clasificar las ondas de acuerdo a su valor K dado por las presiones y sus valores K_1 y K_2 dado por las formas de sus ondas.(figura 3-5)

Redefiniendo la ecuación 3-13 de acuerdo a los valores K_1 y k_2 [7]. El gasto cardiaco es calculado como:

$$CO = \frac{500K}{(K + K_1 + K_2)^3} * (P_s - P_d) \quad (3-14)$$

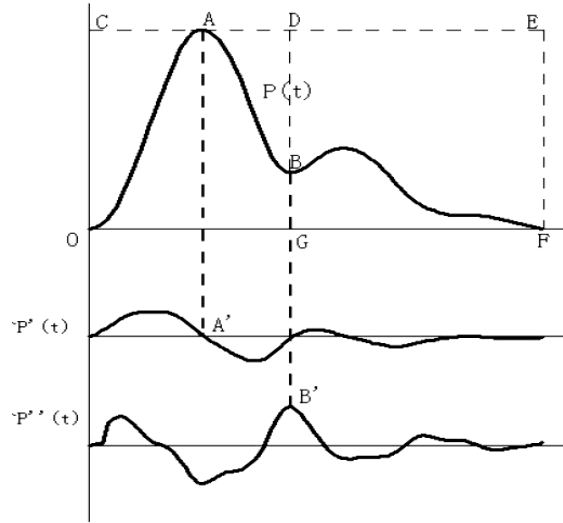


Figura 3-4: División de la onda de pulso por medio de la onda dicrótica [7]

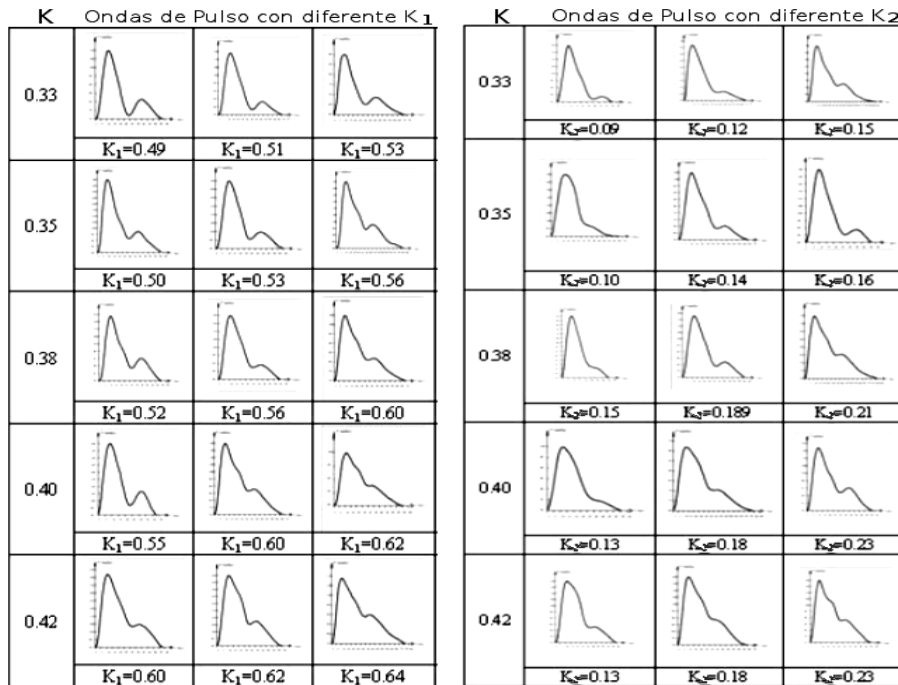


Figura 3-5: Diferentes formas de onda variando K , K_1 y K_2 [7]

3.3. Análisis de datos y resultados

Un total de 32 personas con rango de edad entre 20 a 70 años fueron evaluados por medio de ecocardiografía transtorácica con registros paralelos de ECG y PPG a través del modulo creado en el presente proyecto.

Se procuró escoger personas en estado saludable sin problemas cardiacos considerables o arritmias continuas. El tiempo de medición promedio es de 5 minutos dependiendo de la ventana del paciente (mejor o peor visualización de imagen del corazón por ecocardiografía). Los errores de calculo producidos por manipulación del paciente durante la prueba, ondas de pulso con patrones bruscos y malas detecciones de picos fueron obviados.

Los resultados de CO y SV se logran gracias a los datos de A_o (diámetro del anillo aórtico) y VTI (integral velocidad-tiempo) arrojados por el ecocardiógrafo.

La tabla 3-1 muestra los principales datos para el total de personas monitorizadas.

Para encontrar un valor de K asociado al PWTT se tuvo en cuenta la relación existente entre el BSA (área de superficie corporal) y el gasto cardiaco [1, p. 232]. La Figura 3-6 muestra la correlación entre el BSA y la constante K derivada del gasto cardiaco para el análisis por PWTT.

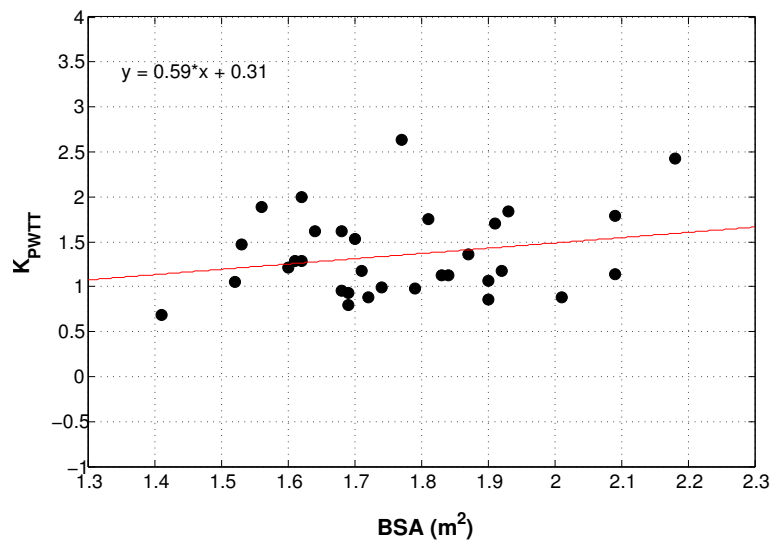


Figura 3-6: Correlación entre el BSA (área de superficie corporal) y la constante K en el método de PWTT

En la tabla 3-2 se puede observar la media y desviación estándar del tiempo de tránsito de pulso para las 32 personas evaluadas, además los datos referentes al área de sección aórtica CSA, integral velocidad-tiempo VTI, frecuencia cardiaca HR, volumen

Parámetro	Valor
No de sujetos	32
Género (<i>M/F</i>)	(17/15)
Edad (<i>Años</i>)	34 ± 18
Peso (<i>Kg</i>)	68,4 ± 11,8
Estatura (<i>cm</i>)	167,5 ± 8,7
HR (Frecuencia cardiaca, <i>latidos/min</i>)	74 ± 12
<i>SpO₂</i> (Saturación de oxígeno, %)	97 ± 2
SBP (Presión Arterial Sistólica, <i>mmHg</i>)	123 ± 17
DBP (Presión Arterial Diastólica, <i>mmHg</i>)	72 ± 13
BSA (Área de la superficie corporal, <i>m²</i>)	1,76 ± 0,18
SV (Volumen Sistólico, <i>mL/ltido</i>)	65,5 ± 14,2
CO (Gasto Cardíaco, <i>L/min</i>)	4,8 ± 1,3

Tabla 3-1: *Parámetros físicos y hemodinámicos de los sujetos de prueba. *Datos representados como media ± desviación estándar*

sistólico SV y gasto cardiaco CO.

Para visualizar los acercamientos a la medida de referencia se ha usado análisis por medio de Bland-Altman, el cual es un método estadístico que permite comparar datos de dos técnicas de medida diferentes. Las gráficas de Bland-Altman muestran la diferencia entre dos técnicas contra sus promedios.

Los resultados permiten determinar:

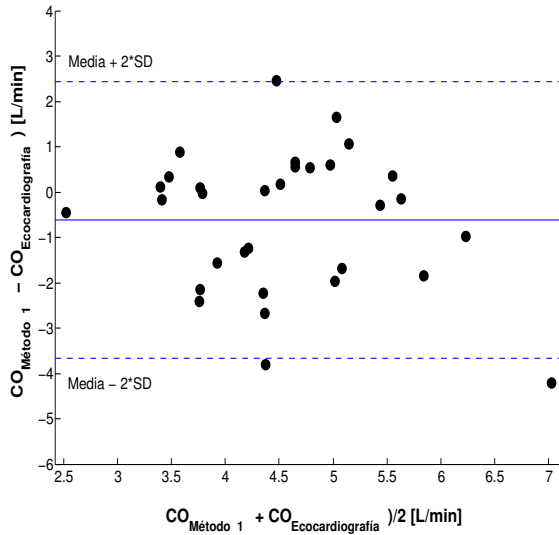
- Sesgo: diferencia promedio. El ideal es cero.
- Precisión: 1 desviación estándar que describe el rango para el 68 % para los puntos de comparación.
- Límite de acuerdo: 2 desviaciones estándar que describen el rango del 95 % de los puntos de comparación.

Mientras la correlación muestra la comparación entre dos medidas diferentes, quizás no sean apropiadas en las medidas de gasto cardiaco, esto se debe a que la correlación es mas usada en medidas de referencia con bajo error. Como el método de referencia usado tiene un rango elevado de error la tendencia se describe mejor con los gráficos de Bland-Altman que con correlación [18].

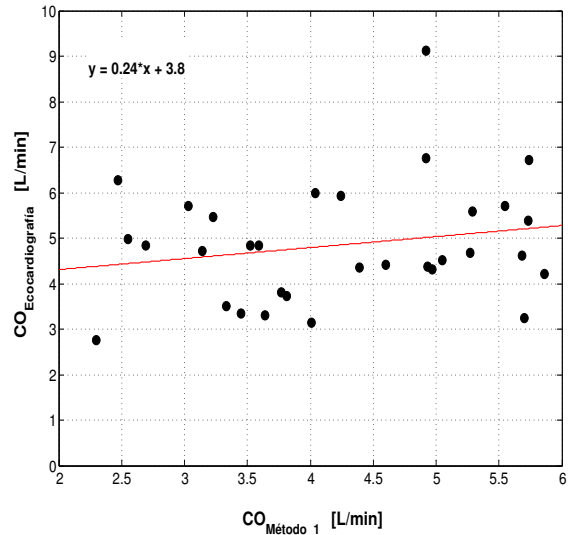
Persona	PWTT (<i>ms</i>)	CSA (<i>cm</i> ²)	VTI (<i>cm</i>)	HR (<i>latidos/min</i>)	SV (<i>mL/latido</i>)	CO (<i>L/min</i>)
1	208,0 ± 6,3	2,83	26,25	63	74,28	4,68
2	242,9 ± 8,0	2,83	24,84	54	70,31	3,80
3	207,9 ± 6,5	3,14	21,10	102	66,28	6,76
4	206,7 ± 5,8	2,83	19,60	67	55,57	3,72
5	192,9 ± 6,2	3,14	23,20	75	72,88	5,47
6	233,0 ± 4,9	3,43	18,50	90	63,46	5,71
7	217,1 ± 4,4	3,39	23,80	69	80,87	5,58
8	210,0 ± 4,0	3,80	19,75	80	75,02	6,00
9	215,1 ± 7,4	2,26	21,08	70	47,66	3,34
10	239,7 ± 5,0	3,14	21,70	71	68,17	4,84
11	251,4 ± 3,6	2,38	17,40	76	41,37	3,14
12	207,0 ± 4,1	2,98	21,65	77	64,51	4,97
13	207,4 ± 5,8	2,77	16,14	74	44,70	3,31
14	206,4 ± 8,0	2,57	18,42	74	47,34	3,50
15	182,3 ± 4,5	2,83	20,73	77	58,69	4,52
16	235,8 ± 4,9	3,46	18,11	69	62,69	4,32
17	217,8 ± 6,9	2,89	21,74	77	62,83	4,84
18	197,1 ± 8,1	2,68	23,29	70	62,44	4,37
19	221,7 ± 4,5	2,86	17,89	85	51,19	4,35
20	231,7 ± 8,4	3,33	15,31	54	50,99	2,75
21	209,1 ± 7,5	3,56	24,52	77	87,30	6,72
22	216,6 ± 6,6	4,11	23,15	96	95,14	9,13
23	257,4 ± 7,1	2,48	19,87	66	49,27	3,25
24	227,6 ± 6,3	3,17	24,02	75	76,15	5,71
25	219,4 ± 7,4	3,30	20,10	71	66,34	4,71
26	201,5 ± 6,0	2,63	25,20	73	66,28	4,84
27	236,7 ± 5,9	3,59	29,33	40	105,3	4,21
28	225,1 ± 4,8	3,43	22,60	81	77,53	6,28
29	206,9 ± 5,4	2,57	24,52	94	63,03	5,92
30	235,0 ± 4,4	2,77	22,54	74	62,45	4,62
31	223,4 ± 4,6	2,96	22,17	82	65,62	5,38
32	203,2 ± 9,1	2,63	23,00	73	60,49	4,42

Tabla 3-2: Variables medidas durante las pruebas a cada persona. PWTT(*media ± desviación estándar), CSA (área del anillo aórtico), VTI (integral velocidad-tiempo), HR (frecuencia cardiaca), SV (volumen sistólico) y CO (gasto cardiaco)

Los resultados de estimaciones de gasto cardiaco y volumen sistólico se muestran en las Figuras 3-7 a la 3-12. Las comparaciones estadísticas se hacen mediante gráficos de Bland Altman y gráficos de correlación para cada uno de los métodos; el método 1 con

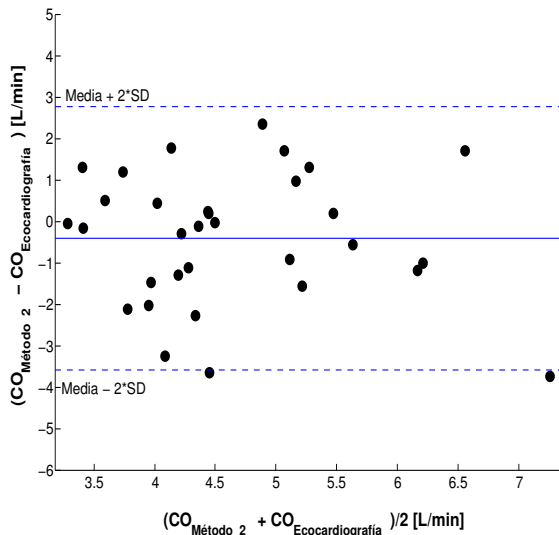


(a)

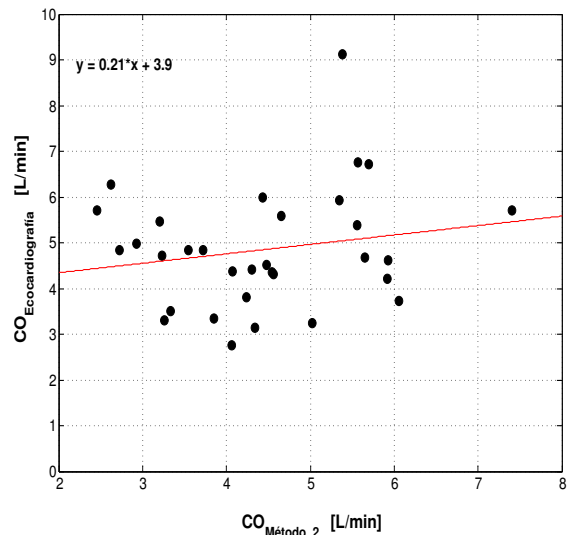


(b)

Figura 3-7: Sesgo y precisión entre gasto cardíaco por método 1 y ecocardiografía. a) Gráfica de Bland y Altman. b) Correlación lineal



(a)

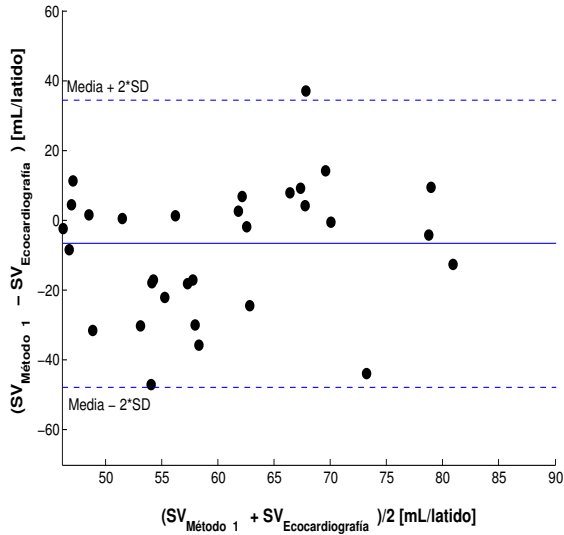


(b)

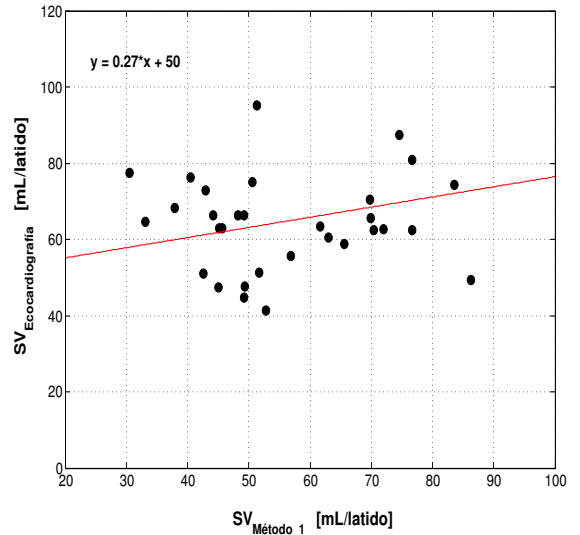
Figura 3-8: Sesgo y precisión entre gasto cardíaco por método 2 y ecocardiografía. a) Gráfica de Bland y Altman. b) Correlación lineal

K, el método 2 con K mejorado K1, K2 y el método 3 con PWTT.

La tabla 3-3 muestra los resultados obtenidos para la comparación de cada uno

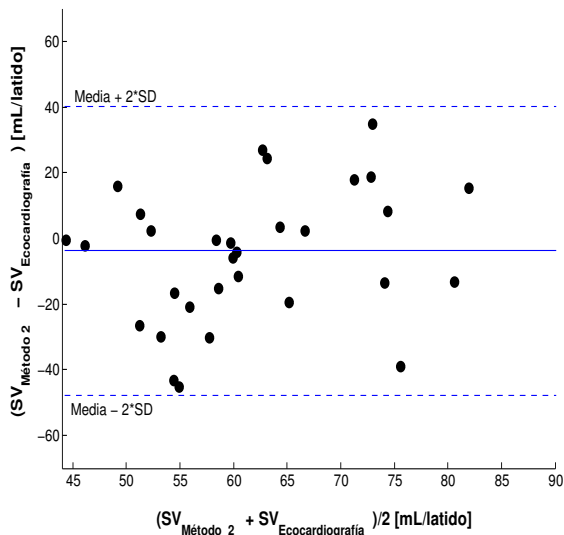


(a)

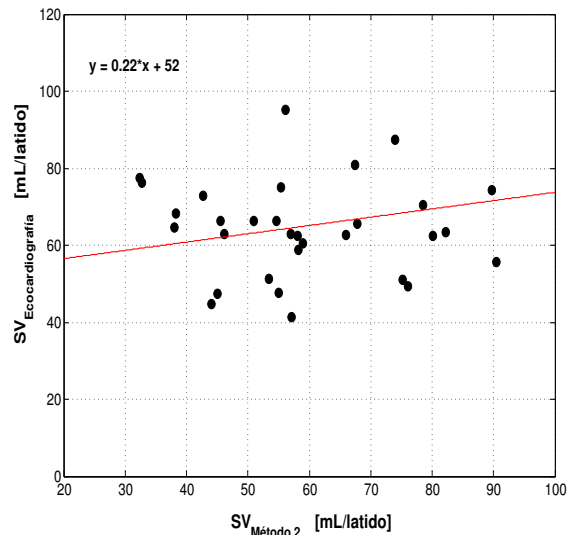


(b)

Figura 3-9: Sesgo y precisión entre volumen sistólico por método 1 y ecocardiografía. a) Gráfica de Bland y Altman. b) Correlación lineal



(a)



(b)

Figura 3-10: Sesgo y precisión entre volumen sistólico por método 2 y ecocardiografía. a) Gráfica de Bland y Altman. b) Correlación lineal

de los métodos con el valor de referencia por ecocardiografía. Los límites de acuerdo superior e inferior hacen referencia a 2 desviaciones estándar o al 95 % de precisión.

Se puede observar que el sesgo en el gasto cardiaco se encuentra cercano al cero

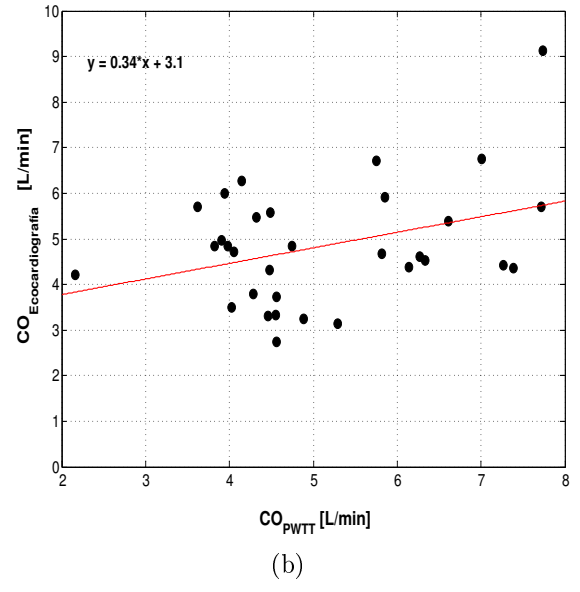
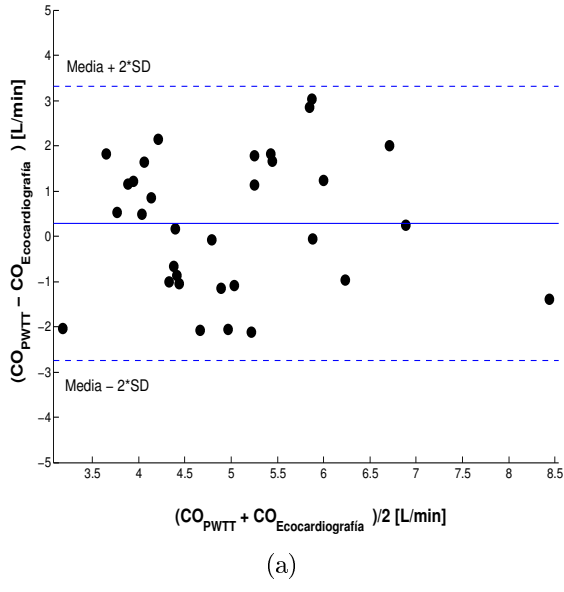


Figura 3-11: Sesgo y precisión entre gasto cardíaco mediante PWTT y ecocardiografía. a) Gráfica de Bland y Altman. b) Correlación lineal

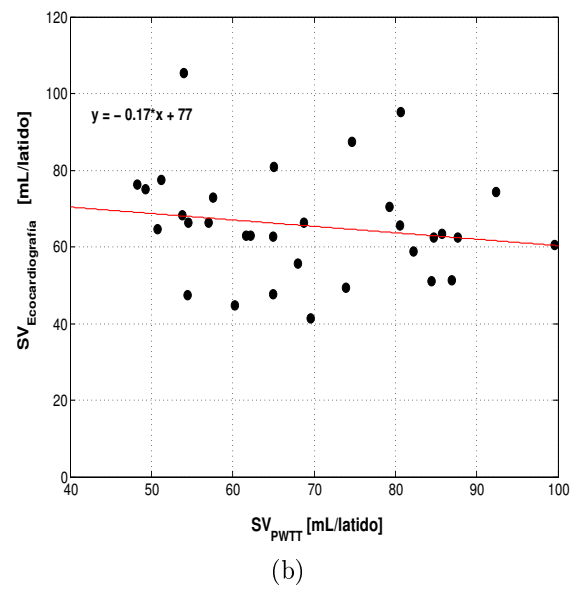
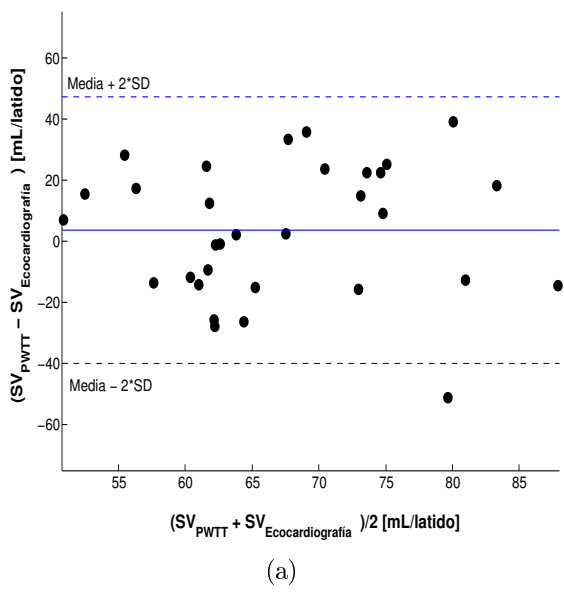


Figura 3-12: Sesgo y precisión entre volumen sistólico mediante PWTT y ecocardiografía. a) Gráfica de Bland y Altman. b) Correlación lineal

evidenciando un error sistemático bajo entre las dos medidas. Por otro lado el valor negativo en los sesgos relativos al método de contorno de pulso indican una sub-estimación del gasto y volumen sistólico mientras el método de tiempo de tránsito de pulso calibra-

	r	p	sesgo	precisión	Lim.(+)	Lim.(-)
CO_{m1}	0,20	0,26	-0,61	1,53	2,43	-3,66
CO_{m2}	0,19	0,30	-0,40	1,59	2,77	-3,59
CO_{PWTT}	0,36	0,04	0,28	1,52	3,32	-2,75
SV_{m1}	0,41	0,02	-6,64	20,55	34,47	-47,75
SV_{m2}	0,34	0,06	-3,78	22,01	40,24	-47,80
SV_{PWTT}	0,17	0,35	3,54	21,81	47,17	-40,09

Tabla 3-3: Resumen de correlaciones y gráficos de Bland-Altman. CO en [L/min] y SV en [mL/latido]

do con datos del paciente los sobre-estima. Los porcentajes de error para las pruebas se hallaron dividiendo los límites de acuerdo, entre los promedios de los resultados estimados frente al método de referencia, encontrando un 60 % para PWTT y un 67 % para el análisis de contorno de pulso a un 95 % de precisión, y del 30 % y 34 % a un 68 % de precisión, alejándose del mínimo error clínicamente aceptado para medidas de gasto cardiaco, correspondiente al 30 % a 95 % de precisión.

Este porcentaje alto puede estar relacionado a errores inherentes al método de referencia. En caso de obtener límites de acuerdo cercanos a 1 L/min es posible estar en el rango clínico aceptado. De todas formas el método PWTT muestra una mejor tendencia al valor de referencia.

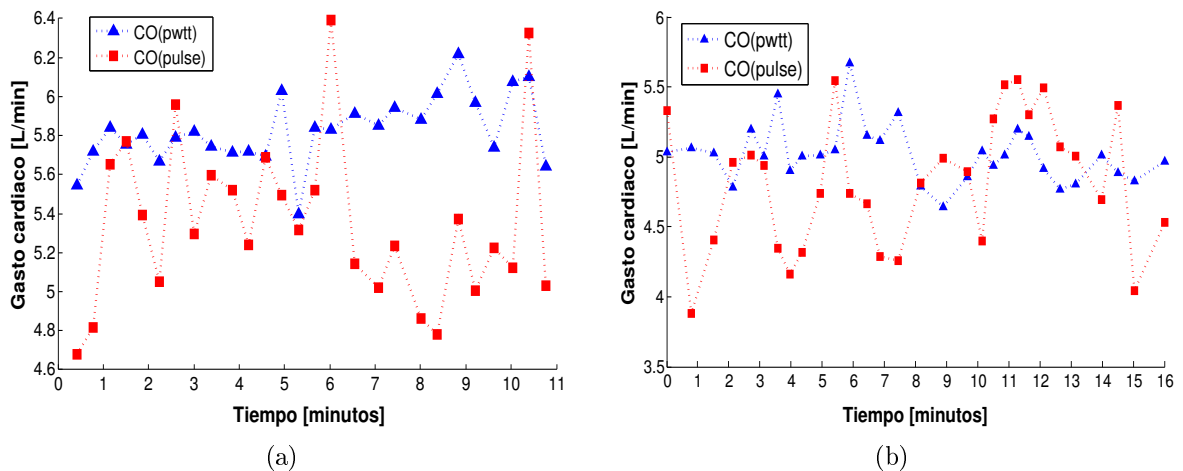


Figura 3-13: Comparación de estabilidad método PWTT vs método de análisis de pulso. a) Sujeto 1. b) Sujeto 2

El método de PWTT calibrado con un valor de gasto inicial tiende a ser mas exacto y preciso al transcurrir de las horas [16], pero en este proyecto no se cuenta con una segunda o tercera medida de gasto en intervalos considerables de tiempo. Sin embargo en la gráfica 3-13 se puede apreciar la estabilidad del método PWTT respecto al de contorno de pulso para un intervalo de tiempo mayor a 10 minutos.

EL comportamiento de los cálculos de gastos para las dos personas es mas estable con el método de PWTT que con el generado por análisis de contorno de pulso, esto es muy importante si se tiene en cuenta la importancia no solo de la exactitud de la medida sino de sus cambios en el tiempo que son un factor determinante a la hora de dar un tratamiento adecuado al paciente.

Capítulo 4

Conclusiones y observaciones

Se realizó la implementación del modulo de pulsioximetría GOLDWEI SP100 junto con el diseño y puesta en marcha de un electrocardiógrafo bipolar sencillo con circuito de pierna derecha, empotrado en una tarjeta PCB junto al microcontrolador, para lograr la sincronización y envío simultaneo de las señales electrocardiográfica y fotople-tismográfica vía serial al PC.

Se utilizó el microcontrolador FREESCALE MCF51QE128 para realizar la comunicación serial por protocolo RS232 de manera simultánea a diferentes tasas de baudios, 4800 baudios en recepción y 19200 para transmisión, con el fin de mantener una tasa de muestreo de 480 muestras/segundo para las señales de interés.

Se evaluaron los métodos de estimación de gasto cardiaco y volumen sistólico a través de análisis de contorno de onda de pulso y tiempo de transito de onda, contrastando estos métodos con valores reales obtenidos por ecocardiografía Doppler transtorácica.

Por medio del software Matlab se diseñó una interfaz gráfica amigable con el usuario, que permite la visualización del porcentaje de saturación de oxigeno en la sangre, las ondas electrocardiográfica y fotople-tismográfica del usuario, además del cálculo de

frecuencia cardiaca, tiempo de tránsito de onda de pulso, gasto cardiaco y volumen sistólico por los dos métodos propuestos.

Se realizó el procesamiento digital de las señales electrocardiográficas y fotople-tismográficas realizando filtrados en el dominio del tiempo y frecuencia, además de implementar eficientes algoritmos de detección de picos, valles, dicróticas y puntos de ascenso, además de criterios de selección de ondas que disminuyen el error en la estimación de gasto cardiaco y volumen sistólico.

Se validó la capacidad del electrocardiógrafo y el pulsioxímetro para proporcionar valores correctos de tiempo de transito de onda de pulso en contraste con el equipo PowerLab 16/30 del Hospital Universitario de Santander.

La sensibilidad del pulsioxímetro al movimiento y factores ambientales son causantes de una señal con contornos irregulares, llevando a errores en los cálculos de índices en onda de pulso.

El cálculo de volumen sistólico y gasto cardiaco a través de ecocardiografía se ve afectado considerablemente por medidas erróneas del anillo aórtico, integral de velocidad y malas prácticas en la medida de presión arterial.

Si bien el método de tiempo de tránsito de onda de pulso es mas aceptable con valores iniciales de gasto cardiaco, la imposibilidad de obtener varios valores de gasto en el tiempo, impidieron obtener mejores resultados, aun así se demostró la estabilidad del método frente al análisis de contorno de pulso usando únicamente calibración por área de superficie corporal BSA.

4.1. Recomendaciones para trabajos futuros

Mejorar el desempeño del ECG, enfocando su diseño al rechazo de ruidos en diversas bandas de frecuencia, poseer un excelente rechazo en modo común y acople de impedancias, descartando así diversas interferencias de tipo electrónico y biológico.

La ecocardiografía transtorácica no es el “*gold standard*” en la determinación del gasto cardiaco y el volumen sistólico. La medida tomada por este método depende en gran parte de la precisión y experiencia del operador del equipo, pues se deben tomar sobre un monitor de computadora manejando manualmente cursores y delineando áreas, todo lo cual conlleva a un error considerable en la medición. La recomendación es contrastar el método con la termodilución de bolo frío, o con catéter arterial pulmonar los cuales son métodos mas aceptados y así poder obtener mejores resultados en los valores estimados.

El dispositivo puede ser usado en otro tipo de investigaciones que requieran cualquiera de las siguientes variables fisiológicas tratadas en el presente proyecto y su relación en tiempo real, tales como estudios relacionados con la arteriosclerosis, hipoxia, anestesia, reanimación, estimación de presión arterial no invasiva entre otros.

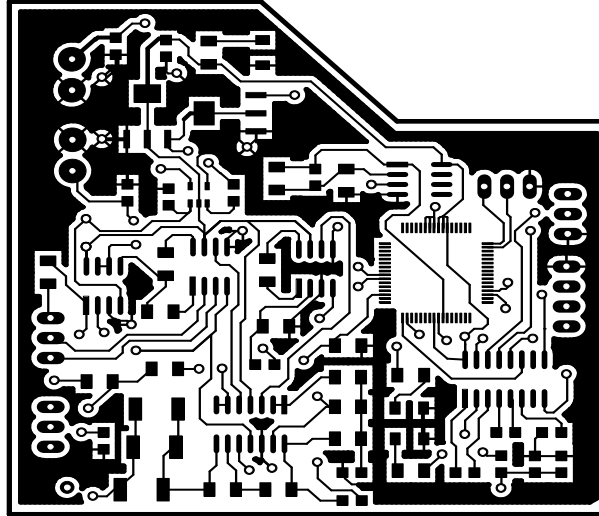
Es posible utilizar el dispositivos para estudiar variaciones del gasto cardiaco y el volumen sistólico en largos periodos de tiempo, además de los cambios que se puedan presentar en diferentes tipos de poblaciones.

Bibliografía

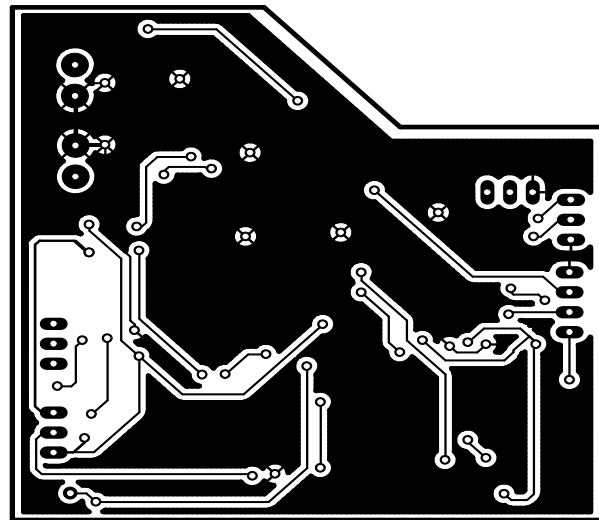
- [1] A. Guyton and J. Hall, *Fisiología Medica*. Elsevier, 2006.
- [2] Best, Taylor, Dvorkin, and D. Cardinali, *Best& Taylor: Bases fisiológicas de la práctica médica*. Panamericana, 2003.
- [3] M. Vannan, R. Lang, H. Rakowski, and A. Tajik, *Atlas of Echocardiography*. Current Medicine Group, 2005.
- [4] N. V. Thakor, J. G. Webster, and W. J. Tompkins, “Estimation of QRS complex power spectra for design of a QRS filter.,” *IEEE transactions on bio-medical engineering*, vol. 31, no. 11, pp. 702–6, 1984.
- [5] N. K. CORPORATION, “A Novel Technology to Non-invasively Measure Continuous Cardiac Output from ECG and SpO2 - Performance Report,”
- [6] J. Chandler, E. Cooke, C. Petersen, W. Karlen, N. Froese, J. Lim, and J. Ansermino, “Pulse oximeter plethysmograph variation and its relationship to the arterial waveform in mechanically ventilated children,” *Journal of Clinical Monitoring and Computing*, vol. 26, pp. 145–151, 2012. 10.1007/s10877-012-9347-z.
- [7] J. Li, L. Yang, S. Zhang, and Y. Yang, “Computation of Cardiac Output by Pulse Wave Contour,” *Stroke*, pp. 1088–1090, 2007.
- [8] L. Wang, “Noninvasive cardiac output estimation using a novel photoplethysmogram index,” *Conf Proc IEEE Eng Med Biol Soc. 2009*, pp. 1746–1749, 2009.

- [9] B. De La Quintana, *Monitorizacion En Anestesia, Medicina De Urgencias Y Cuidados Intensivos*. Elsevier - Health Sciences Division, 2004.
- [10] J. Allen, “Photoplethysmography and its application in clinical physiological measurement,” *Physiological Measurement*, vol. 28, no. 3, p. R1, 2007.
- [11] D. P.-S. Louise Akin, “Health library, cirugias y procedimientos, electrocardiograma.” <http://healthlibrary.ochsner.org/Spanish/TestsProcedures/92,P09310>, 04 2010.
- [12] R. E. Klabunde, *Cardiovascular Physiology Concepts*. Lippincott Williams & Wilkins, 2004.
- [13] M. Chakravarthy, “Cardiac output, have we found the gold standard?,” *Ann Card Anaesth*, vol. 11, no. 1, pp. 1–2.
- [14] J. Gámiz and P. L., *Electrocardiografía de Holter: Bases prácticas y aplicaciones clínicas*. Ediciones Norma, 1983.
- [15] a. M. Weissler, W. S. Harris, and C. D. Schoenfeld, “Systolic Time Intervals in Heart Failure in Man,” *Circulation*, vol. 37, no. 2, pp. 149–159, 1968.
- [16] Y. Sugo, T. Ukawa, S. Takeda, H. Ishihara, T. Kazama, and J. Takeda, “A novel continuous cardiac output monitor based on pulse wave transit time.,” *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, vol. 2010, no. C, pp. 2853–6, 2010.
- [17] H. Ishihara and H. Okawa, “A new non-invasive continuous cardiac output trend solely utilizong routinecardiovascular monitors comparison with the continuous,” no. 2004, pp. 313–320, 2005.
- [18] E. L. LLC, “Statistical Methodology for Comparison of Cardiac Output Measurements Bland-Altman Analysis. What is a Bland-Altman analysis ?,” pp. 2–3.

Apéndices



Capa superior



Capa inferior

Figura -2: Layout capa superior e inferior

Programación del Microcontrolador

MCF51QE128

En este apéndice se describe la configuración de cada uno de los módulos del microcontrolador MCF51QE128 utilizados en el proyecto y su respectivo código de Codewarrior. En una segunda parte se describen las funciones de usuario con sus respectivos códigos y finalmente el código principal.

Configuración de módulos del microcontrolador

Configuración inicial de la CPU del microcontrolador

Módulo *CPU* >Clock settings >**Internal bus clock:** 25.165824 MHz

Configuración inicial del temporizador

Módulo *TPM1* >Settings >Clock settings >**Prescaler:** 1, **Modulo counter:** 251, **Period:** 10.013 us

```
/* ### Init_TPM init code */
/* TPM1SC: TOF=0,TOIE=0,CPWMS=0,CLKSB=0,CLKSA=0,PS2=0,PS1=0,PS0=0 */
TPM1SC = 0; /* Stop and reset counter */
TPM1MOD = 251U; /* Period value setting */
(void)(TPM1SC == 0); /* Overflow int. flag clearing (first part) */
/* TPM1SC: TOF=0,TOIE=0,CPWMS=0,CLKSB=0,CLKSA=1,PS2=0,PS1=0,PS0=0 */
TPM1SC = 8; /* Int. flag clearing (2nd part) and timer
control register setting */
```

Configuración del modulo de Comunicación serial para Transmisión

Módulo *SCI1* >Settings >Clock settings >
Baud rate divisor: 82, **Baud rate:** 19181.268 baud,
Settings >
TxD pin direction: Output, **Data format:** 8 bits,**Parity:** None
Pins >
RxD pin allocation: Enabled, **RxD pin:** PTB0_KBI1P4_RxD1,
TxD pin allocation: Enabled, **TxD pin:** PTB1_KBI1P5_TxD1
Initialization >
Transmitter: Enabled, **Receiver:** Disabled

```
/* ### Init_SCI1 (Tx) init code */
/* SCI1C2: TIE=0,TCIE=0,RIE=0,ILIE=0,TE=0,RE=0,RWU=0,SBK=0 */
```

```

SCI1C2 = 0;          /* Disable the SCI1 module */
(void)(SCI1S1 == 0); /* Dummy read of the SCI1S1 register to clear flags */
(void)(SCI1D == 0); /* Dummy read of the SCI1D register to clear flags */
/* SCI1S2: LBKDIF=1,RXEDGIF=1,RXINV=0,RWUID=0,BRK13=0,LBKDE=0,RAF=0 */
SCI1S2 = 192;
/* SCI1BDH: LBKDIE=0,RXEDGIE=0,SBR12=0,SBR11=0,SBR10=0,SBR9=0,SBR8=0 */
SCI1BDH = 0;
/* SCI1BDL: SBR7=0,SBR6=1,SBR5=0,SBR4=1,SBR3=0,SBR2=0,SBR1=1,SBR0=0 */
SCI1BDL = 82;
/* SCI1C1: LOOPS=0,SCISWAI=0,RSRC=0,M=0,WAKE=0,ILT=1,PE=0,PT=0 */
SCI1C1 = 4;
/* SCI1C3: R8=0,T8=0,TXDIR=1,TXINV=0,ORIE=0,NEIE=0,FEIE=0,PEIE=0 */
SCI1C3 = 32;
/* SCI1C2: TIE=0,TCIE=0,RIE=0,ILIE=0,TE=1,RE=0,RWU=0,SBK=0 */
SCI1C2 = 8;

```

Configuración inicial del modulo de Comunicación serial para Recepción

Módulo *SCI2* >Settings >Clock settings >

Baud rate divisor: 328, Baud rate: 4795.317 baud

Settings >

TxD pin direction: Input, Data format: 8 bits, Parity: Odd

Pins >

RxD pin allocation: Enabled, RxD pin: PTC6_RGPIO14_RxD2,

TxD pin allocation: Enabled, TxD pin: PTC7_RGPIO15_TxD2

Initialization >

Transmitter: Disabled, Receiver: Enabled

```

/* ### Init_SCI2 (Rx) init code */
/* SCI2C2: TIE=0,TCIE=0,RIE=0,ILIE=0,TE=0,RE=0,RWU=0,SBK=0 */
SCI2C2 = 0;          /* Disable the SCI2 module */
(void)(SCI2S1 == 0); /* Dummy read of the SCI2S1 register to clear flags */
(void)(SCI2D == 0); /* Dummy read of the SCI2D register to clear flags */
/* SCI2S2: LBKDIF=1,RXEDGIF=1,RXINV=0,RWUID=0,BRK13=0,LBKDE=0,RAF=0 */
SCI2S2 = 192;
/* SCI2BDH: LBKDIE=0,RXEDGIE=0,SBR12=0,SBR11=0,SBR10=0,SBR9=0,SBR8=1 */
SCI2BDH = 1;
/* SCI2BDL: SBR7=0,SBR6=1,SBR5=0,SBR4=0,SBR3=1,SBR2=0,SBR1=0,SBR0=0 */
SCI2BDL = 72;
/* SCI2C1: LOOPS=0,SCISWAI=0,RSRC=0,M=0,WAKE=0,ILT=1,PE=1,PT=1 */
SCI2C1 = 7;
/* SCI2C3: R8=0,T8=0,TXDIR=1,TXINV=0,ORIE=0,NEIE=0,FEIE=0,PEIE=0 */
SCI2C3 = 32;
/* SCI2C2: TIE=0,TCIE=0,RIE=0,ILIE=0,TE=0,RE=1,RWU=0,SBK=0 */
SCI2C2 = 4;

```

Configuración inicial del Convertidor Analógico Digital (ADC)

Módulo *ADC* >Settings >Clock settings >

Prescaler: 8, Frequency: 3145.728 kHz, Conversion time: 6.56 us (152.520 kHz)

Settings >

Conversion mode: Single conversion, Result data format: 8-bit right,

Low power mode: Enabled

Pins >

ADC Input Pins: 2

Input Pin0 >

Pin: PTF4_AD14

Input Pin1 >

Pin: PTF1_AD11

Initialization >

Initial channel select: Channel 14

```

/* ### Init_ADC init code */
/* APCTL2: ADPC15=0,ADPC14=0,ADPC13=0,ADPC12=0,ADPC11=0,ADPC10=0,ADPC9=0,
  ADPC8=0 */
APCTL2 = 0;
/* ADCCFG: ADLPC=1,ADIV1=1,ADIVO=1,ADLSMP=0,MODE1=0,MODE0=0,ADICLK1=0,
  ADICLK0=0 */
ADCCFG = 224;
/* ADCCV: ADCV11=0,ADCV10=0,ADCV9=0,ADCV8=0,ADCV7=0,ADCV6=0,ADCV5=0,ADCV4=0,
  ADCV3=0,ADCV2=0,ADCV1=0,ADCV0=0 */
ADCCV = 0U;
/* ADCSC2: ADACT=0,ADTRG=0,ACFE=0,ACFGT=0 */
ADCSC2 = 0; //pag.223
/* ADCSC1: COCO=0,AIEN=0,ADC0=0,ADCH4=0,ADCH3=1,ADCH2=1,ADCH1=1,ADCH0=0 */
ADCSC1 = 14; //pag.222
/* ### */

```

Configuración inicial de LEADS

```

/* ### Init PTCDD - inicializacion LEDs */
PTCDD_PTCDD0 = 1; //Led Amarillo
PTCDD_PTCDD1 = 1; //Led Verde
/* ### */

```

Descripción de las funciones de usuario

Función Temporizador o Retardo

```

/**Funcion TEMPORIZADOR con TPM en us**/
//temp=251 corresponde ~10us (10.013us); max_temp=65535
void TPMus(unsigned short temp){
  TPM1MOD = temp;
  TPM1SC = 0x08; //inic TPM con prescaler=0
  TPM1CNT = 0;
  while (TPM1SC_TOF == 0) { }
  TPM1SC = 0x00; //desactiva TPM
} // end TPMus

```

Esta función tiene por objetivo hacer un retardo cuando sea llamada, el argumento es la variable *temp* que actualiza el registro *TPM1MOD* y dependiendo de su valor será el tiempo de retardo. Para un *temp=251* se tendrá un retardo de 10 us.

Función para leer el ADC

```

/**Funcion LEER ADC**/
byte ReadADC(byte channel) {
  ADCSC1_ADCH=channel;
  while (!(ADCSC1_COCO)){ } // Waits until ADC conversion is completed
  return ADCRL;
} // end LEER ADC

```

La función *ReadADC* realiza una sola lectura de un canal del ADC cada vez que se ejecuta (el proceso de una lectura dura 6.56 us); el canal es seleccionado colocando en el argumento el número correspondiente. Si se ejecuta *ReadADC(14)* se está leyendo una muestra del valor actual del Electrocardiograma; en cambio, si se ejecuta *ReadADC(11)* se leerá una muestra de la onda Pletismográfica.

Funciones para leer un Byte del SCI2 o enviar un Byte por el SCI1

```

/**Funciones ESCRIBIR (1) y LEER (2) serial**/
char SendChar(char msj1) {
    while(!SCI1S1_TDRE){ }
    SCI1D = msj1; }
char ReadChar() {
    if (SCI2S1_RDRF) // 1st half of RDRF clear procedure
        msj2 = SCI2D; // 2nd half of RDRF clear procedure
    SCI2C2_RE = 1; // enable Rx
    while(!SCI2S1_RDRF) {};
    msj2 = SCI2D; // get recieved character
    return (char) msj2;
} // end SendChar y ReadChar

```

El modulo de comunicación serial para recepción *SCI2* fue configurado para funcionar a un *Baud rate* de 4800 baudios y el de transmisión *SCI1* a un *Baud rate* de 19200 baudios.

Según el protocolo de envío serial, se espera que el registro *SCI1S1_TDRE* está activado para colocar en el registro *SCI1D* el byte de datos a enviar, definido por la variable *msj1* como argumento de la función *SendChar(msj1)*.

Para la recepción de un byte por medio del *SCI2* se hacen los dos pasos del protocolo de limpieza y se habilita la recepción haciendo el registro *SCI2C2_RE = 1*. Se debe hacer una espera mientras el registro *SCI2S1_RDRF* sea cero. Cuando este registro se hace uno, se recibe el byte y se devuelve como salida.

Descripción del código principal

```

232. void main(void) {
233.     init_micro(); //Inicializacion del micro
234.
235. /* include your code here */
236.
237.     p=12; //Adquisicion inicial del valor de SpO2
238.     for(k=0;k<10;k++){
239.         mensaje[k]=(byte)ReadChar();
240.         m = (unsigned int) mensaje[k];
241.         if(m-127>0 && p>11)
242.             p=k;
243.     }
244.     data_SP=mensaje[p+4]; //SpO2 value

```

En la línea 232 se define la función principal *main()* y en seguida se llama la función *init_micro()*, definida en la línea 10, para inicializar el microcontrolador. Con el bucle *for* entre líneas 236 y 240 se hace una lectura inicial de dos paquetes de 5 bytes generados por el modulo GOLDWEI SP100 (*tabla 2-1*). Cada Byte del paquete tiene una longitud de 11 bits: 1 de start, 8 de datos, 1 de stop, 1 de paridad más 5 bits de IDLE entre Bytes) y recibidos en el SCI2 a 4800 baudios, esta lectura inicial tendría una duración aproximada de 33.33 ms. Mientras se hace la lectura se busca el Byte 1 del paquete, que tendrá un bit de sincronismo igual a 1 en la última posición o bit 7. Una vez sea encontrado, se guarda el índice *k* en la variable *p* y de acuerdo al orden de los bytes en el paquete, se busca el Byte 5, que contiene el valor de SpO2 generado por el modulo GOLDWEI SP100, y se guarda en la variable *data_SP*.

```

246. for(;;) {
247.

```

```

248. //Envio de los 4 paquetes de datos (10 bits c/u)
249. SendChar((char)0xF8);
250. //TPMus(11710); //IDLE 2 bits (104us - 6.56us de ADC)
251.
252. data_ECG=ReadADC(14); //(6.56us)
253. SendChar(data_ECG);
254. //TPMus(11710); //IDLE 2 bits (104us - 6.56us de ADC)
255.
256. data_PPG=ReadADC(11); //(6.56us)
257. SendChar(data_PPG);
258. //TPMus(12500); //IDLE 2 bits (104us)
259.
260. SendChar(data_SP); //SpO2 value
261. //TPMus(12450); //IDLE 2 bits (104us - 0.5us de refresco)
262.
263. PTCD_PTCD1 = ~PTCD_PTCD1;

```

En la línea de código 246 comienza el bucle infinito, que hará que el microcontrolador funcione continuamente sin detenerse. De la línea 249 a 261 se escribe el código necesario para enviar paquetes de 4 Bytes a través del módulo de comunicación serial SCI1. Los 4 Bytes contienen respectivamente: Un carácter bandera, una muestra de la señal de ECG, una muestra de la señal de PPG y un valor de Saturación de oxígeno SpO2. Una representación del paquete enviado y el contenido de cada Byte es mostrada en la figura 2-8 Al final del envío de cada paquete se ha puesto el código `PTCD_PTCD1 = not(PTCD_PTCD1)`, como en la línea 263. El registro `PTCD_PTCD1` controla un LED, y hace que cambie su estado cada vez que se envía un paquete, esto permite la medición de la frecuencia de muestreo en laboratorio.

Las líneas 250, 254, 258 y 261 se encuentran comentadas, y dan la posibilidad al usuario del uso de retardos para disminuir la Tasa de muestreo de envío. Habilitando estos retardos se añaden bits de *IDLE* o inactividad a la transmisión serial, aumentando el número de bits enviados en cada paquete, lo que acrecienta el tiempo entre muestras.

```

265. //Refresco del valor de SpO2 (Lectura SCI2)
266. p=12;
267. for(k=0;k<10;k++){
268.
269.     if (SCI2S1_RDRF) // 1st half of RDRF clear procedure
270.         msj2 = SCI2D; // 2nd half of RDRF clear procedure
271.     SCI2C2_RE = 1; // enable Rx
272.
273.     while(!SCI2S1_RDRF) {
274.         //Envio de los 4 paquetes de datos (10 bits c/u)
275.         SendChar((char)0xF8);
276.         //TPMus(11710); //IDLE 2 bits (104us - 6.56us de ADC)
277.
278.         data_ECG=ReadADC(14); //(6.56us)
279.         SendChar(data_ECG);
280.         //TPMus(11710); //IDLE 2 bits (104us - 6.56us de ADC)
281.
282.         data_PPG=ReadADC(11); //(6.56us)
283.         SendChar(data_PPG);
284.         //TPMus(12500); //IDLE 2 bits (104us)
285.
286.         data_SP=data_SP;
287.         SendChar(data_SP); //SpO2 value
288.         //TPMus(12450); //IDLE 2 bits (104us - 0.5us de refresco)
289.
290.         PTCD_PTCD1 = ~PTCD_PTCD1;
291.     }
292.
293.     msj2 = SCI2D; // get recieved character

```

El envío de cada paquete de 4 Bytes tarda aproximadamente 2 ms, y la recepción de dos paquetes de 5 Bytes generados por el modulo SP100 tarda aproximadamente 33.33 ms. El microcontrolador debe enviar continuamente al PC los paquetes de 4 Bytes para mantener la Tasa de muestreo; por lo cual no es posible esperar los 33.33 ms que tarda la lectura de los dos paquetes. Es necesario durante el protocolo de recepción de cada Byte realizar el envío de los 4 paquetes, aprovechando que el Baud rate de envío (19200 baudios) es 4 veces mayor que el de recepción (4800). Las líneas 269 a 293 contienen el protocolo de recepción del modulo *SCI2*, al igual que se hizo en la función *ReadChar()* definida en las líneas 220 a 227, con la excepción que el bucle *while* vacío que se tenía en la línea 224: `while(!SCI2S1_RDRF)`; (que representa el tiempo de espera de lectura de 1 Byte, o 11 bits de recepción, total = 2.33 ms), ahora contiene las líneas 274 a 290 que representan el envío de los 4 paquetes, al igual que en las líneas 248 a 263.

```

293.     msj2 = SCI2D; // get recieved character
294.
295.     mensaje[k]= (byte) msj2;
296.     m = (unsigned int) mensaje[k];
297.     if(m-127>0 && p>11)
298.         p=k;
299.     }
300.
301.     data_SP=mensaje[p+4]; //Valor SpO2 actualizado
302.
303.     if(data_SP==0x7F) PTCD_PTCD0 = 1;
304.     else PTCD_PTCD0 = 0;
305.
306. } /* loop forever */

```

En conclusión, en lugar de esperar la recepción de los dos paquetes completos para iniciar el envío, se aprovecha el vacío que hay en la espera para enviar los 4 paquetes sin interrupción. Después de finalizar la lectura el registro *SCI2S1_RDRF* se activa, y procede a leer el registro de datos *SCI2D* almacenando su valor en la variable *msj2* (Línea 293). Finalmente en las líneas 301 a 304 se controla el estado del LED *PTCD_PTCD0* indicando si se está o no enviando el valor de *data_SP* especificado. La línea 306 contiene el cierre del bucle *for(;;)* continuo que contiene el código principal y que se inició en la línea 246.

Código de Procesado y visualización en Matlab

En este apéndice son descritos los códigos necesarios para la ejecución del programa en Matlab. El código principal se encuentra en el archivo *project_esCO.m* y está vinculado con la interfaz gráfica *project_esCO.fig*, este es descrito en la sección 4.1.

Los demás códigos hechos por los autores y que son llamados por el código principal se encuentran en los archivos *CualPuerto.m*, *config.m*, *iniciar_puerto.m*, *imaxecg.m*, *imaxppg.m*, *i3dppg.m*, *idicppg.m*, *copulse.m* y *esCOvar.m* y son mostrados en la sección 4.1.

Descripción del código principal: *project_esCO.m*

```
1. function varargout = project_esCO(varargin)
2. global close
3. close = 0;
4.
5. % Company: UIS
6. % Engineers: Carlos A. Calderon, Carlos A. Borrás
7. % Project Managers: MsC. Carlos Andres Nino, PhD. Daniel Alfonso Sierra
8. % Create Date: Oct/2012
9. % Project Name: Estimacion de Volumen Sistolico y Gasto Cardiaco
10. % mediante la implementacion de un Pulsioximetro
11. % usando Analisis de contorno de Pulso
12. % Target Devices: GOLDWEI SP100 SpO2 Module & EKG Board with Freescale
    MCF51QE128
13. % Description: Codigo para la adquisicion de las senales a traves de
14. % puerto USB (RS232), procesamiento digital de estas (filtrado Butterworth
15. % y de ventana en frecuencia, ademas deteccion de picos y otros puntos),
16. % calculo de frecuencia cardiaca HR, tiempo de transito de onda de pulso
17. % PWTT, volumen de eyeccion sistolico SV y gasto cardiaco CO.
18.
19.
20. % Begin initialization code - DO NOT EDIT
    {...}
37. % End initialization code - DO NOT EDIT
```

En estas líneas de código se define la función principal que lleva el nombre del código: *project_esCO()* y en líneas siguientes se describe brevemente. Las líneas 20 a 37

son creadas automáticamente por Matlab para interactuar con la interfaz gráfica *project_esCO.fig* y no se muestran aquí.

Inicialización del programa e interfaz gráfica

```
40. % --- Executes just before project_esCO is made visible.
41. function project_esCO_OpeningFcn(hObject, eventdata, handles, varargin)
    {...}
49.
50. gu=imread('usb.png');
51. [fil,col,cap]=size(gu);
52. x=ceil(fil/70); y=ceil(col/70);
53. im1=gu(1:x:end,1:y:end,:);
54. set(handles.puerto,'CData',im1);
    {...}
80. set(gcf,'CurrentAxes',handles.axes3)
81. IM = imread('titulo.png');
82. imshow(IM);
    {...}
111.
112. t=datestr(now,'dd/mmm/yyyy HH:MM');
113. set(handles.fecha,'string',t)
114. set(handles.calculando,'String','');
115. set(handles.storebutton,'Value',0)
```

Las líneas 40 a 111 contienen el código de inicialización de la interfaz gráfica, en el cual se asignan las figuras que contendrán los botones y cuadros, como se muestra en las líneas 50 a 54 y 80 a 82 respectivamente. Las líneas 112 a 115 inicializan algunos valores de texto como la fecha y hora actual y el estado del botón *storebutton*.

```
118. % --- Executes during object creation, after setting all properties.
119. function axes1_CreateFcn(hObject, eventdata, handles)
120. function axes2_CreateFcn(hObject, eventdata, handles)
121. function axes3_CreateFcn(hObject, eventdata, handles)
122. function axes4_CreateFcn(hObject, eventdata, handles)
123. function axes5_CreateFcn(hObject, eventdata, handles)
124. function axes6_CreateFcn(hObject, eventdata, handles)
125. function axes7_CreateFcn(hObject, eventdata, handles)
126.
127. % --- Executes during object creation, after setting all properties.
128. function fecha_CreateFcn(hObject, eventdata, handles)
129. function spo_CreateFcn(hObject, eventdata, handles)
130. function gasto_pwtt_CreateFcn(hObject, eventdata, handles)
131. function stroke_pwtt_CreateFcn(hObject, eventdata, handles)
132. function calculando_CreateFcn(hObject, eventdata, handles)
133.
134. function varargout = project_esCO_OutputFcn(hObject, eventdata, handles)
135. % Get default command line output from handles structure
136. varargout{1} = handles.output;
137.
138. % --- Executes when user attempts to close figure1.
139. function figure1_CloseRequestFcn(hObject, eventdata, handles)
140.
141. global close
142.
143. opc = questdlg('Esta seguro de salir del programa?', 'SALIR', 'Si', 'No',
    'No');
```

```

144. if strcmp(opc, 'No')
145.     return
146. end
147. close = 1;
148.
149. load varconfigpuert
150. fclose(SerPIC);
151. delete(SerPIC);
152. clear SerPIC
153.
154. delete(hObject);
155. display('Programa terminado')

```

En las líneas 118 a 132 contiene las funciones que el GUIDE de Matlab crea automáticamente al crear un objeto. Luego aparece la función de salida del código principal y las líneas 138 a 155 contienen el código para cerrar el programa desde la interfaz gráfica.

```

157. % --- Executes when figure1 is resized.
158. function figure1_ResizeFcn(hObject, eventdata, handles)
159.
160. % --- Executes on button press in puerto.
161. function puerto_Callback(hObject, eventdata, handles)
162. CualPuerto
163.
164. % --- Executes on button press in config.
165. function config_Callback(hObject, eventdata, handles)
166. config

```

En las líneas 162 y 166 se llaman y ejecutan los archivos *CualPuerto.m* y *config.m* los cuales se ejecutan al presionar los botones *Port* y *Config* respectivamente. Cada uno de ellos abre una ventana adicional con su respectiva acción, como se muestra en la sección 2.2.7. Los códigos *CualPuerto.m* y *config.m* se muestran en la sección 4.1 este apéndice.

■ Función del botón REC

```

168. %% --- Executes on button press in REC.
169. function storebutton_Callback(hObject, eventdata, handles)
170. dir = strcat(cd, '/userdata/');
171. nombre = importdata(strcat(dir, 'info_USERNAME.mat'));
172. info = importdata(strcat(dir, 'info_', nombre, '.mat'));
173.
174. savef0 = strcat(dir, nombre, '_time.mat');
175. savef1 = strcat(dir, nombre, '_PPG.mat');
176. savef2 = strcat(dir, nombre, '_ECG.mat');
177. savef3 = strcat(dir, nombre, '_time2.mat');
178. savef4 = strcat(dir, nombre, '_Sp_HR_PWTT.mat');
179. savef5 = strcat(dir, nombre, '_K12_SV_CO.mat');
180. savefn = strcat(dir, 'data_', nombre, '.mat');
181.
182. T1 = now;
183. tim = datestr(T1-T1, 'HH:MM');
184. tim = strcat('RecordTime:', tim);
185. set(handles.fecha, 'string', tim)
186. set(handles.calculando, 'String', '(Grabando)', 'ForegroundColor', 'w');
187.
188. PPG = []; ECG = []; Sp_HR_PWTT = []; K12_SV_CO = []; datoin = [];
189. while (get(hObject, 'Value'))

```

```

190.     [PPGs, ECGs, Sp_HR_PWTTs, K12_SV_COs, datoins] = Iniciar_Callback(
        hObject, eventdata, handles, 1);
191.     PPG = [PPG, PPGs];
192.     ECG = [ECG, ECGs];
193.     Sp_HR_PWTT = [Sp_HR_PWTT, Sp_HR_PWTTs];
194.     K12_SV_CO = [K12_SV_CO, K12_SV_COs];
195.     datoin = [datoin datoins];
196.
197.     T2 = now;
198.     tim = datestr(T2-T1,'MM:SS');
199.     tim = strcat('RecordTime:',tim);
200.     set(handles.fecha,'string',tim)
201. end
202.
203. Lsig = length(ECG)-1;
204. time = (0:1/481:Lsig/481);
205. time2 = (0:1500/481:Lsig/481);
206.
207. save(savef0, 'time');
208. save(savef1, 'PPG');
209. save(savef2, 'ECG');
210. save(savef3, 'time2');
211. save(savef4, 'Sp_HR_PWTT');
212. save(savef5, 'K12_SV_CO');
213. save(savefn, 'datoin');

```

Las líneas 169 a 245 contienen todo el código ejecutado al presionar el botón *REC* de la interfaz gráfica, que a continuación se describirá brevemente. Primeramente el código importa de la carpeta */userdata/* (ubicada en la carpeta raíz) los archivos *info_USERNAME.mat* y *info_nombre.mat* que contienen respectivamente el nombre y los datos almacenados del usuario puestos en la ventana *Config*.

En las líneas 174 a 180 se crean variables tipo *string* con los nombres de los archivos a almacenar, que posteriormente se guardan con los resultados en las líneas 207 a 213. Las líneas 182 a 186 muestran el tiempo de grabación en pantalla y el estado *Grabando*. Posteriormente en las líneas 188 a 205 se crean los vectores vacíos *PPG*, *ECG*, *Sp_HR_PWTT*, *K12_SV_CO* y *datoin* que contendrán respectivamente los resultados de las señales que describen sus nombres, estos se obtienen como salidas de la función principal *Iniciar_callback()*.

```

215. %Creacion se secuencia de archivos para evitar sobrescritura de datos
216. savenombre = strcat(cd,'/userdata/', 'info_USERNAME.mat');
217.
218. Serie = str2double(nombre(end));
219. if isnan(Serie)
220.     nombre = strcat(nombre,'_1');
221.     save(savenombre, 'nombre');
222. else
223.     Serie2 = str2double(nombre(end-1));
224.     if isnan(Serie2) && Serie == 9 %es 9
225.         nn = num2str(10);
226.         nombre = nombre(1:end-2);
227.         nombre = strcat(nombre,'_',nn);
228.     elseif isnan(Serie2) %es 1 a 8
229.         nn = num2str(Serie+1);
230.         nombre = nombre(1:end-2);
231.         nombre = strcat(nombre,'_',nn);
232.     elseif Serie == 9 %es 19 o 29...
233.         nombre = nombre(1:end-3);
234.         nn = num2str(10*(Serie2+1));

```

```

235.         nombre = strcat(nombre, '_', nn);
236.     else %es 11 a 18 o 21 a 28 o...
237.         nombre = nombre(1:end-3);
238.         nn = num2str(10*(Serie2)+Serie+1);
239.         nombre = strcat(nombre, '_', nn);
240.     end
241. end
242. save(savenombre, 'nombre');
243. saveinfo = strcat(dir, 'info_', nombre, '.mat');
244. save(saveinfo, 'info');
245. set(handles.calculando, 'String', '');

```

Como continuación del código anterior, se muestra en las líneas 215 a 245 el código que permite tomar múltiples muestras de un mismo paciente sin sobrescribir las anteriores, agregando un índice al nombre del usuario almacenado en *info_USERNAME.mat* sin modificar la demás información. Es posible guardar hasta 99 tomas para el mismo paciente sin sobrescritura.

■ Función del botón STOP

```

248. %% --- Executes on button press in STOP.
249. function parar_Callback(hObject, eventdata, handles, times)
250. % hObject    handle to parar (see GCBO)
251. % eventdata  reserved - to be defined in a future version of MATLAB
252. % handles    structure with handles and user data (see GUIDATA)
253. global close
254. if close == 1
255.     return
256. end
257.
258. while (get(hObject, 'Value'))
259.     pause(0.3);
260.     if close == 1
261.         return
262.     end
263.     set(handles.text_distancias, 'Visible', 'off')
264.     set(handles.pushdistancia, 'Visible', 'on')
265. end

```

El botón STOP tiene como función detener la muestra de resultados de la interfaz gráfica, además para evitar errores se debe presionar antes de cerrar el programa. En las líneas 264 se habilita el botón *pushdistancia* para medir distancia entre puntos de la gráfica en pantalla, este procedimiento se describe en el código a continuación.

■ Función para medir distancia entre puntos

```

267. %% --- Executes on button press in Medir distancias.
268. function pushdistancia_Callback(hObject, eventdata, handles)
269. % hObject    handle to pushdistancia (see GCBO)
270. % eventdata  reserved - to be defined in a future version of MATLAB
271. % handles    structure with handles and user data (see GUIDATA)
272.
273. cm=colormap(hsv(256));
274. rcm = round(255*rand)+1; %color del punto aleatorio
275.
276. [gx1,gy1] = ginput(1);
277. p1 = impoint(gca, gx1, gy1);
278. setColor(p1, cm(rcm, :))

```

```

279.
280. [gx2,gy2] = ginput(1);
281. p2 = impoint(gca, gx2, gy2);
282. setColor(p2,cm(rcm,:))
283.
284. dist = roundn(abs(gx2-gx1),-3);
285. ampl = roundn(abs(gy2-gy1),-3);
286. Dist = strcat(num2str(dist),' s');
287. set(handles.distancia,'string',Dist)
288. set(handles.amplitud,'string',num2str(ampl))
289. return

```

La función *pushdistancia_Callback()* permite mediante el comando *ginput()* de Matlab definir dos puntos en las gráficas que se encuentren en pantalla para que el usuario pueda medir tiempo o amplitudes entre los puntos. Cada par de puntos saldrá de un color aleatorio diferente y las medidas se mostrarán en el panel ubicado en la parte inferior derecha de la interfaz gráfica (ver figura 2-31).

Algoritmo principal

■ Función del botón START

```

292. %% --- Executes on button press in START.
293. function [PPG, ECG, Sp_HR_PWTT, K12_SV_CO, datoin] = Iniciar_Callback(
    hObject, eventdata, handles, times)
294.
295. global close
296. if close == 1
297.     return
298. end
299.
300. %importe de los datos del usuario
301. dir = strcat(cd,'/userdata/');
302. nombre = importdata(strcat(dir,'info_USERNAME.mat'));
303. info = importdata(strcat(dir,'info_',nombre,'.mat'));
304.
305. Weigth = info(2);
306. Heigth = info(3);
307. Ps = info(4);
308. Pd = info(5);
309.
310. %inicializacion de variables
311. Count = 0; HR = NaN; PWTT = NaN; SpO = NaN; HRv = []; PWTTv = [];
    COv = [];
312.
313. if ~exist('times','var')
314.     times = inf;
315.     set(handles.calculando,'String','(Por favor espere)',
        'ForegroundColor','w');
316.     iniciar_puerto();
317. end
318.
319. set(handles.pushdistancia,'Visible','off')
320. set(handles.text_distancias,'Visible','on')

```

En la línea 293 se define la función principal del algoritmo, llamada *Iniciar_Callback()*. Sus argumentos de entrada los pone por defecto Matlab, excepto el argumento *times*, que determina el número de veces que se correrá la función, cuando no se especifica,

será *inf* (infinito) por defecto. Los argumentos de salida son vectores que contienen los resultados de cada una de las medidas calculadas por el algoritmo para un paso (cada paso de la función captura un número de datos *ldata* que por defecto es 6000).

Al entrar en la función se importan los datos del usuario del vector *info* y se guardan en las variables respectivas (líneas 305 a 308). En la línea 311 se inicializan variables necesarias durante el proceso. En las líneas 313 a 317 se verifica la existencia de la variable *times*, sino está definida significa que el código se corre por primera vez y se define igual a infinito permitiendo que el programa corra indefinidamente.

Finalmente se llama la función *iniciar_puerto()*, la cual contiene toda la configuración necesaria para iniciar el puerto serial en la ubicación seleccionada por el usuario al presionar el botón *Port*. Esta función es descrita en la sección 4.1 de este apéndice.

■ Bucle de ejecución continua

```

322. %bucle que contiene el algoritmo principal
323. while (get(hObject,'Value') == 1 && times > 0)
324. % tic
325.     times = times - 1;
326.     Count = Count + 1;
327.     if mod(Count+1,3)==0
328.         t = datestr(now,'dd/mmm/yyyy HH:MM');
329.         set(handles.fecha,'string',t)
330.     end

```

Se define el bucle *while* para ejecutar continuamente el algoritmo principal mientras que las condiciones de llamado de la función sea verdadera y que la variable *times* tenga un valor mayor que cero. La variable *times* controla el número de ejecuciones del algoritmo principal, disminuyendo su valor en 1 por cada paso dentro del bucle *while*. La variable *Count* es usada para contar el número de pasos del algoritmo principal y controlar la actualización cada 5 pasos de la fecha y la hora que no requiere modificarse a cada paso del *while*.

```

332.     ldata = 6000; %cantidad de datos recopilados
333.     mps = 481; %muestras por segundo
334.     lsignal=round(ldata/4);
335.
336.     ecg=zeros(1,lsignal);
337.     ppg=zeros(1,lsignal);
338.     spo2=zeros(1,lsignal);
339.
340.     %% Adquisicion de los datos
341.     load varconfigpuert
342.     datoin = fread(SerPIC,ldata+4,'uint8');
343. %     DATA = importdata(strcat(dir,'data_P10_CarlosCalderon.mat'));
344. %     datoin = DATA(:,Count);

```

La variable *ldata* define la cantidad de datos que recogerá el puerto, debe ser múltiplo de 4 (debido a que los paquetes recogidos son de 4 bytes); La variable *mps* representa el número de muestras por segundo dado por el microcontrolador, en este caso igual a 481 (véase la sección 2.1.3).

Para adquirir los datos se carga la configuración del puerto en la variable *varconfigpuert.mat* que fue creada por la función *iniciar_puerto()* y a través de la función

fread() se recogen los datos en formato 'uint8' y se almacenan en la variable *datoin*. Se recoge una cantidad de *ldata+4* datos para evitar desbordamiento de los bucles y pérdida de muestras. Las líneas 343 y 344 se dejan como opción en caso de que se quieran visualizar datos guardados.

■ Extracción de las señales del vector de datos

```

346.     c = 0;
347.     for i = 1 : ldata
348.         if datoin(i) == 248 && (datoin(i+3) > 49 && datoin(i+3) < 101 ||
           datoin(i+3) == 127)
349.             c = c+1;
350.             ecg(c)=datoin(i+1);
351.             ppg(c)=datoin(i+2);
352.             spo2(c)=datoin(i+3);
353.         end
354.     end
355.
356.     lsignal = c;
357.     t = (0:1/mps:(lsignal-1)/mps);
358.     PPG = (ppg/255);
359.     ECG = (ecg/103.03); %factor de correccion 2.475/255
360.     SpO = spo2(lsignal-1);

```

La variable *c* indica el índice de las señales *ecg(c)*, *ppg(c)* y *spo2(c)*, que aumenta cada vez que comprueba la validez del valor de SpO2 almacenado en la posición *datoin(i+3)* y se encuentra la bandera *datoin(i) == 248* enviada por el microcontrolador en el paquete de 4 bits. Una vez se encontrada la bandera, el dato siguiente *datoin(i+1)* corresponde a una muestra de la onda de ECG, el siguiente corresponde a una de PPG y el ultimo a un valor de saturación de oxígeno. Se define además el vector de tiempo *t* para fines de ploteo y se normalizan los valores de las señales de ECG y PPG.

■ Aplicación de filtros a las señales

```

362.     %% Filtrado de las ondas
363.     fNorm = 53/(mps/2);
364.     [b,a] = butter(13, fNorm, 'low');
365.     ECG = filtfilt(b, a, ECG);
366.
367.     fNorm = 53/(mps/2);
368.     [b,a] = butter(13, fNorm, 'low');
369.     PPG = filtfilt(b, a, PPG);
370.
371.     %Filtrado pasa altas para deteccion de picos de ECG
372.     fECG = fft(ECG);
373.     lf = length(fECG);
374.     fECG(1 : round(lf*0.5/(mps/2))) = 0;
375.     fECG(end - round(lf*0.5/(mps/2)) : end)=0;
376.     ECGc = real(ifft(fECG));
377.
378.     %Filtrado pasa altas para deteccion de picos de PPG
379.     fPPG = fft(PPG);
380.     lf = length(fPPG);
381.     fPPG(1 : round(lf*0.1/(mps/2))) = 0;
382.     fPPG(end - round(lf*0.1/(mps/2)) : end) = 0;
383.     PPGc = real(ifft(fPPG));

```

Las líneas 362 a 383 contienen el código para hacer el filtrado digital Butterworth y de ventana en frecuencia como se había explicado en la sección 2.2.1 donde se explica el filtrado aplicado.

■ Detección de máximos, mínimos y puntos clave de las ondas

```

385.     %% Deteccion de maximos, minimos y puntos clave
386.     [nmaxECG, imaxECG, ampECG] = imaxecg(ECGc);
387.     if Sp0 ~= 127
388.         [nmaxPPG, imaxPPG, nminPPG, iminPPG] = imaxppg(PPGc);
389.         [n3dPPG, i3dPPG] = i3dppg(PPGc, nminPPG, iminPPG);
390.         [ndicPPG, idicPPG] = idicppg(PPGc, imaxPPG, iminPPG);
391.     else
392.         nmaxPPG=0; nminPPG=0; n3dPPG=0; ndicPPG=0;
393.         imaxPPG=[]; iminPPG=[]; i3dPPG=[]; idicPPG=[];
394.     end

```

El procedimiento y los resultados de la detección de máximos, mínimos, puntos dicróticos y de ascenso son mostrados en las secciones 2.2.2 y 2.2.3. Las funciones *imaxecg()*, *imaxppg()*, *i3dppg()* y *idicppg()* son mostradas en la sección 4.1.

■ Determinación de áreas bajo la curva PPG

El código mostrado a continuación permite encontrar el área bajo la curva de la onda de PPG antes y después del punto dicrótico, además del calculo de los valores de K , K_1 y K_2 como fue mostrado en la sección 3.2.

```

396.     %% Determinacion de areas bajo la curva PPG
397.     PPGn = PPGc - min(PPGc); %colocada sobre cero
398.     PPGn = PPGn/max(PPGn); %normalizacion de la onda
399.
400.     npoint = sort([ndicPPG nminPPG nmaxPPG]);
401.     b1 = []; b2 = []; b3 = [];
402.     valid = 0;
403.     for i = 1:ndicPPG
404.         [a1] = find(imaxPPG < idicPPG(i));
405.         b1 = [b1 imaxPPG(a1(end))];
406.         [a2] = find(iminPPG > idicPPG(i));
407.         b3 = [b3 iminPPG(a2(1))];
408.         b2 = [b2 idicPPG(i)];
409.     end
410.     bb = [b1' b2' b3'];
411.     [hbb,lbb] = size(bb);
412.     tol = 0.07;
413.
414.     if hbb == 2 || hbb == 3
415.         if PPGn(bb(2,2)) < PPGn(bb(1,2))+tol && PPGn(bb(2,2)) >
           PPGn(bb(1,2))-tol
416.             valid = 1;
417.         end
418.     elseif hbb > 3
419.         if ( PPGn(bb(2,2)) < PPGn(bb(1,2))+tol && PPGn(bb(2,2)) >
           PPGn(bb(1,2))-tol ) && ( PPGn(bb(2,2)) < PPGn(bb(3,2))+tol && ...
           PPGn(bb(2,2)) > PPGn(bb(3,2))-tol )
420.             valid = 1; end
421.         if valid == 0
422.             if ( PPGn(bb(3,2)) < PPGn(bb(2,2))+tol && PPGn(bb(3,2)) >
           PPGn(bb(2,2))-tol ) && ( PPGn(bb(3,2)) < PPGn(bb(4,2))+tol...

```

```

424.         && PPGn(bb(3,2)) > PPGn(bb(4,2))-tol )
425.             valid = 2; end
426.     end
427. end
428.
429. K = NaN; K1 = NaN; K2 = NaN;
430.
431. if valid > 0
432.     v = valid - 1;
433.     PPGz = PPGn(bb(1+v,3):bb(2+v,3)); %una onda completa
434.     d1 = bb(2+v,2)-bb(1+v,3);
435.     d2 = bb(2+v,3)-bb(2+v,2);
436.     mxPPGz = max(PPGz);
437.     A1 = sum(PPGz(1:d1)); %area de minimo a dicrotica
438.     B1 = mxPPGz*(d1); %area primer rectangulo
439.     K1 = A1/B1;
440.     A2 = sum(PPGz(d1+1:end)); %area de dicrotica a minimo
441.     B2 = mxPPGz*(d2); %area segundo rectangulo
442.     K2 = A2/B2;
443.
444.     Pm = Pd + (Ps-Pd)*sum(PPGz)/(d1+d2-1);
445.     K = (Pm-Pd)/(Ps-Pd);
446. end

```

■ Cálculo de frecuencia cardiaca

```

448.     %% Calculo del Heart Rate
449.     if nmaxECG > 3 && nmaxECG < 11 && ampECG > 0.2
450.         dimaxECG = diff(imaxECG);
451.         HRvector = sort(dimaxECG);
452.         HRvector = HRvector(1,2:nmaxECG-2);
453.         HR = round(60*mps/mean(HRvector));
454.     elseif nmaxPPG + nminPPG > 3 && SpO ~= 127
455.         dimax = diff(imaxPPG);
456.         dimin = diff(iminPPG);
457.         HRvector = sort([dimax dimin]);
458.         npeak=nmaxPPG+nminPPG;
459.         HRvector = HRvector(1,2:npeak-2);
460.         HR = round(60*mps/mean(HRvector));
461.     elseif nmaxECG == 3 || nmaxECG == 2
462.         HRvector = diff(imaxECG);
463.         HR = round(60*mps/mean(HRvector));
464.     end

```

La frecuencia cardiaca se calcula promediando la distancia entre picos de la onda de ECG si se cumple la condición de la línea 449, en otro caso se calcula hallando el promedio de las distancias entre máximos y entre mínimos de la onda de PPG al cumplirse la condición de la línea 454, en caso contrario no se calcula y el valor a la salida del programa será *NaN* (no es un numero).

■ Cálculo de tiempo de transito de onda de pulso

```

466.     %% Calculo del PWTT
467.     if nmaxECG > 3 && n3dPPG > 3
468.         if imaxECG(1) < i3dPPG(1)
469.             PWTT1 = (i3dPPG(2)-imaxECG(2))/mps;
470.             PWTT2 = (i3dPPG(3)-imaxECG(3))/mps;
471.         else
472.             PWTT1 = (i3dPPG(3)-imaxECG(2))/mps;
473.             PWTT2 = (i3dPPG(4)-imaxECG(3))/mps;

```

```

474.         end
475.         d1 = abs(PWTT-PWTT1); d2 = abs(PWTT-PWTT2);
476.         if d1 > d2
477.             PWTT = PWTT2;
478.         else
479.             PWTT = PWTT1;
480.         end
481.
482.     elseif nmaxECG > 2 && n3dPPG > 2
483.         if imaxECG(1) < i3dPPG(1)
484.             PWTT = (i3dPPG(2)-imaxECG(2))/mps;
485.         else
486.             PWTT = (i3dPPG(3)-imaxECG(2))/mps;
487.         end
488.     end

```

Para calcular el tiempo de transito de la onda de pulso (PWTT) se verifica que haya un mínimo de tres máximos de la onda de ECG y tres puntos de subida de la onda de PPG detectados. El PWTT se calcula como el tiempo entre el pico R del complejo QRS de la onda de ECG y el punto de subida de la onda pletismográfica PPG (véase sección 3.1). En la línea 468 se comprueba cual de los dos picos aparece primero, si aparece primero un máximo de ECG se calcula el tiempo con las ecuaciones de las líneas 469 y 470, de lo contrario se hace con las ecuaciones de las líneas 472 y 473. Por ultimo entre los dos resultados se escoge el valor que tenga menor variabilidad con respecto al valor anterior.

■ Estimación de volumen sistólico y gasto cardiaco

A continuación se muestra el código utilizado para estimar el volumen sistólico y el gasto cardiaco a partir de los coeficientes K encontrados anteriormente y el tiempo de transito de onda de pulso. La estimación se hace promediando siete valores validos de gasto cardiaco para el método de contorno de onda de pulso y promediando siete valores de PWTT y aplicando su formula correspondiente para el método por tiempo de transito.

```

491.     %% Estimacion del Volume sistolico y Gasto cardiaco
492.     SV = [NaN; NaN; NaN];
493.     CO = [NaN; NaN];
494.     esCO = NaN;
495.
496.     if ~isnan(K1)
497.         [co_pulse2,co_pulse1] = copulse(Pd,Ps,K,K1,K2);
498.         CO = [co_pulse2; co_pulse1]/1000;
499.         COv = [COv CO];
500.         lCOv = length(COv);
501.         if lCOv >=3
502.             set(handles.calculando,'String','(Calculando)')
503.             set(handles.calculando,'ForegroundColor',[0.9, 0.4, 0]);
504.         end
505.         if lCOv == 7
506.
507.             lHRv = length(HRv);
508.             if lHRv >= 6
509.                 HRv = sort(HRv);
510.                 elim = round(lHRv/4-1);
511.                 HRm = mean(HRv(elim+1:end-elim));
512.                 HRv = [];
513.             else
514.                 HRm = mean(HRv);

```

```

515.         HRv = [];
516.     end
517.
518.     COa = sort(COv(1,:));
519.     CO1a = sort(COv(2,:));
520.     COa = mean(COa(3:6));
521.     CO1a = mean(CO1a(3:6));
522.     SV(1) = 1000*COa/HRm;
523.     SV(2) = 1000*CO1a/HRm;
524.
525.     if ~isnan(CO1a)
526.         set(handles.calculando,'String','(Resultados)',
             'ForegroundColor','r')
527.         set(handles.gasto_pulse,'string',roundn(CO1a,-3));
528.         set(handles.stroke_pulse,'string',roundn(SV(2),-2));
529.     end
530.
531.     COv = [];
532.
533.     lPWTTv = length(PWTTv);
534.     if lPWTTv >= 6
535.         PWTTv = sort(PWTTv);
536.         elim = round(lPWTTv/4-1);
537.         PWTTm = mean(PWTTv(elim+1:end-elim));
538.         PWTTv = [];
539.     else
540.         PWTTm = mean(PWTTv);
541.         PWTTv = [];
542.     end
543.
544.     BSA = 0.2024 * (Height/100)^0.725 * (Weight)^0.425;
545.
546.     [beta] = esCOvar(PWTTm, Ps, Pd, BSA, 0);
547.     [beta, k_esCO, esSV] = esCOvar(PWTTm, Ps, Pd, BSA, beta);
548.
549.     SV(3) = esSV;
550.     esCO = (esSV * HRm)/1000;
551.
552.     if ~isnan(esCO)
553.         set(handles.gasto_pwtt,'string',roundn(esCO,-3));
554.         set(handles.stroke_pwtt,'string',roundn(esSV,-2));
555.     end
556.
557.     end
558. end
559.
560. K12_SV_CO = [K; K1; K2; SV; CO; esCO]

```

- **Adecuación y envío de valores validos de frecuencia cardiaca y saturación de oxígeno**

Los valores de frecuencia cardiaca y saturación de oxígeno obtenidos anteriormente deben ser adecuados para evitar mostrar errores en el programa, con este fin se implementa el código mostrado a continuación:

```

%% Adecuacion y envio de valores validos de HR y SpO2
if SpO == 127 || SpO < 51
    set(handles.spo,'string','--');
elseif SpO == 100
    SpO = 99;
    set(handles.spo,'string',SpO);
else

```

```

        set(handles.spo,'string',Sp0);
    end

    if isnan(HR) == 1 || HR > 220
        set(handles.rate,'string','--');
    else
        set(handles.rate,'string',HR);
        HRv = [HRv HR];
    end

    if isnan(PWTT) == 1 || (PWTT > 0.35 || PWTT < 0.1)
        set(handles.pwtt_value,'string','--');
    else
        set(handles.pwtt_value,'string',roundn(PWTT*1000,-1));
        PWTTv = [PWTTv PWTT];
    end

    Sp_HR_PWTT = [Sp0; HR; PWTT];

```

■ Ploteo de las gráficas - Fin del algoritmo principal

```

589.     %Para ploteo de los puntos en PPG
590. %     Idic = zeros(1,length(PPG)); minimos = Idic; maximos = Idic;
591. %     for i=1:ndicPPG
592. %         Idic(idicPPG(i)) = PPGc(idicPPG(i));
593. %     end
594. %     for i=1:nmaxPPG
595. %         maximos(imaxPPG(i)) = PPGc(imaxPPG(i));
596. %     end
597. %     for i=1:nminPPG
598. %         minimos(iminPPG(i)) = PPGc(iminPPG(i));
599. %     end
600. %     Idic(Idic==0) = NaN; maximos(maximos==0) = NaN;
        minimos(minimos==0) = NaN;

601.
602. %% Ploteo de las graficas
603. plot(handles.axes1,t,PPG,'LineWidth',1.5);
604. %     plot(handles.axes1,t,PPGc,t,Idic,'or',t,minimos,'og',t,maximos,
        'og','LineWidth',1.5);
605. set(handles.axes1,'Xlim',[0,lsignal/mps])
606. set(handles.axes1,'XGrid','on','YGrid','on','Color',[0,0,0])
607. set(handles.axes1,'XColor',[1 1 1], 'YColor',[1 1 1])
608.
609. plot(handles.axes2,t,ECG,'r','LineWidth',1.5)
610. set(handles.axes2,'Xlim',[0,lsignal/mps],'Ylim',[0,1])
611. set(handles.axes2,'XGrid','on','YGrid','on','Color',[0,0,0])
612. set(handles.axes2,'XColor',[1 1 1], 'YColor',[1 1 1])
613.
614. pause(1.5)
615. if close == 1
616.     return
617. end
618. % toc
619. end

```

Finalmente se procede al ploteo de las señales de PPG y ECG contra el eje temporal en la interfaz gráfica. Opcionalmente las líneas 590 a 600 se pueden habilitar junto con la línea 604 para visualizar en la gráfica de PPG los puntos correspondientes a máximos, mínimos y puntos dicróticos de la onda. En las líneas 614 a 617 se hace una pequeña pause que permite el ploteo de las gráficas y se pregunta por la variable *close* en caso que el usuario haya cerrado el programa. La línea 618 muestra el comando *toc*

que junto con la línea 324 (comando *tic*) se pueden utilizar para medir el tiempo de procesamiento del código principal. En la línea 619 se cierra el *while()* que fue abierto en la línea 323 y que contiene el algoritmo principal.

Códigos adicionales

▪ *CualPuerto.m*

La función *CualPuerto()* está vinculada con la archivo *CualPuerto.fig* y es llamada cuando se oprime el botón *PORT* (véase la figura 2-31). La mayor parte del código es creado automáticamente por Matlab, solo se mostrará a continuación las líneas editadas por los autores:

```
% --- Executes on selection change in listado.
function listado_Callback(hObject, eventdata, handles)
puerto = get(handles.listado,'value');
save valpuerto puerto
pause(0.05)
close CualPuerto
```

▪ *config.m*

La función *config()* está vinculada con la archivo *config.fig* y es llamada cuando se oprime el botón *CONFIG* permitiendo introducir los datos del usuario. Solo se muestran a continuación las líneas editadas por los autores:

```
%% --- Executes on button press in Guardar.
function Guardar_Callback(hObject, eventdata, handles)
% hObject      handle to Guardar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

nombre = get(handles.nombre,'String');

dir = strcat(cd,'/userdata/');
saveinfo = strcat(dir,'info_',nombre,'.mat');
savenombre = strcat(dir,'info_USERNAME.mat');

edad = str2double(get(handles.edad,'String'));
peso = str2double(get(handles.peso,'String'));
estatura = str2double(get(handles.estatura,'String'));
sist = str2double(get(handles.sistolica,'String'));
dias = str2double(get(handles.diastolica,'String'));

info = [edad,peso,estatura,sist,dias];
save(savenombre, 'nombre');
save(saveinfo, 'info');

pause(0.1)
figure1_CloseRequestFcn(config);
```

▪ *iniciar_puerto.m*

```

%funcion para inicializar el puerto serial
function iniciar_puerto()

load varconfigpuert
fclose(SerPIC);
delete(SerPIC)
clear SerPIC

load valpuerto
switch puerto
    case 1
        ttt = 'COM1';
    case 2
        ttt = 'COM2';
    case 3
        ttt = 'COM3';
    case 4
        ttt = 'COM4';
    case 5
        ttt = 'COM5';
    case 6
        ttt = 'COM6';
    case 7
        ttt = 'COM7';
    case 8
        ttt = 'COM8';
    case 9
        ttt = 'COM9';
    case 10
        ttt = 'COM10';
end

SerPIC = serial(ttt);
set(SerPIC,'BaudRate',19200);
set(SerPIC,'DataBits',8);
set(SerPIC,'Parity','none');
set(SerPIC,'StopBits',1);
set(SerPIC,'FlowControl','none');
set(SerPIC,'ReadAsyncMode','Continuous');
set(SerPIC,'Inputbuffersize',18012);
fopen(SerPIC);
save varconfigpuert SerPIC

```

■ *imaxecg.m*

```

function [nmax, imax, amp, m2] = imaxecg(sig)
%funcion para encontrar los indices de los maximos y minimos de una señal
%de tipo ECG

lsignal=length(sig);
media = mean(sig);
mx = max(sig);
mn = min(sig);
m2 = 1.15*(mx+media)/2; %umbral para deteccion de picos
amp = mx-mn;

imax=[];
flg2=0;
nmax = 0;

for j = 1:lsignal

```

```

    if sig(j) > m2 %Dentro de una cresta
        if flg2 == 0
            k21 = j;
            flg2 = 1;
        end
        if j == lsignal %Si llega al final de la señal
            k22 = j;
            flg2 = 0;
            [mx1,imax1] = max(sig(k21:k22));
            imax = [imax imax1+k21-1];
            nmax = nmax+1;
        end
    else
        if flg2 == 1 %Deteccion del maximo en cada cresta
            k22 = j;
            flg2 = 0;
            [mx1,imax1] = max(sig(k21:k22));
            imax = [imax imax1+k21-1];
            nmax = nmax+1;
        end
    end
end
end
end

% ceros = zeros(1,lsignal);
% maximos = ceros;
%
% limax=length(imax);
% for j=1:limax
%     maximos(imax(j))=sig(imax(j));
% end

```

▪ *imaxppg.m*

```

function [nmax, imax, nmin, imin, m1, m2] = imaxppg(sig)
%funcion para encontrar los indices de los maximos y minimos de una señal
%de tipo PPG

lsignal = length(sig);

media = mean(sig);
mx = max(sig);
mn = min(sig);
m2 = 1.10*(mx+media)/2;
m1 = 0.9*(mn+media)/2;

imax=[]; imin=[];
flg2=0; flg1=0;
nmax = 0; nmin = 0;

for j = 1:lsignal
    if sig(j) > m2 %Dentro de una cresta
        if flg2 == 0
            k21 = j;
            flg2 = 1;
        end
        if j == lsignal %Si llega al final de la señal
            k22 = j;
            flg2 = 0;
            if k22 - k21 > 25

```

```

        [mx1,imax1] = max(sig(k21:k22));
        imax = [imax imax1+k21-1];
        nmax = nmax+1;
    end
end
else
    if flg2 == 1 %Deteccion del maximo en cada cresta
        k22 = j;
        flg2 = 0;
        if k22 - k21 > 15
            [mx1,imax1] = max(sig(k21:k22));
            imax = [imax imax1+k21-1];
            nmax = nmax+1;
        end
    end
end
end
if sig(j) < m1 %Dentro de un valle
    if flg1 == 0
        k11 = j;
        flg1 = 1;
    end

    if j == lsignal %Si llega al final de la señal
        k12 = j;
        flg1 = 0;
        if k12 - k11 > 25
            [mn1,imin1] = min(sig(k11:k12));
            imin = [imin imin1+k11-1];
            nmin = nmin+1;
        end
    end
end
else
    if flg1 == 1 %Deteccion del minimo en cada valle
        k12 = j;
        flg1 = 0;
        if k12 - k11 > 15
            [mn1,imin1] = min(sig(k11:k12));
            imin = [imin imin1+k11-1];
            nmin = nmin+1;
        end
    end
end
end
end
end

```

▪ *i3dppg.m*

```

function [n3d, i3d] = i3dppg(sig, nmin, imin)
%funcion para encontrar el punto de ascenso de la onda PPG (30% de derivada)

dsig = diff(sig);
mxdsig = max(dsig);
ldsig = length(dsig);

i3d = [];
n3d = 0;

for j = 1 : nmin
    lim = ldsig-imin(j)-1;
    if lim > 110

```

```

        lim = 110;
    end

    mxdsig = max(dsig(imin(j):imin(j)+lim));
    for k = 1 : lim
        if dsig(imin(j)+k) >= 0.3*mxdsig
            k3 = imin(j)+k;
            n3d = n3d + 1;
            i3d = [i3d k3];
            break;
        end
    end
end
end

```

▪ *idicppg.m*

```

function [ndic, idic] = idicppg(sig, imax, imin)
%funcion para encontrar la dicrotica de la onda PPG

dsig = diff(sig);
ldsig = length(dsig);
idic = [];
ndic = 0;

nmin = length(imin);
nmax = length(imax);

%validacion de maximos y minimos
c = 0;
for i = 1 : nmin
    [a1] = find(imax < imin(i));
    if isempty(a1)
        else
            c = c + 1;
            Imax(c) = imax(a1(end));
            Imin(c) = imin(i);
            aa = length(a1)+1;
            imax = imax(aa:end);
        end
    end

if ~exist('Imax','var')
    return
else
    imax = Imax;
    imin = Imin;
    nmin = length(imin);
end

for j = 1 : nmin
    dic = 0; dic2 = 0; dic3 = 0; %banderas de deteccion
    i_min = imin(j);
    mn = sig(i_min);
    mx = sig(imax(j));
    band1 = mn+0.10*(mx-mn);
    band2 = mn+0.90*(mx-mn);
    for k = imax(j)+10:i_min-10
        if dsig(k-1) >= 0 %criterio derivada cero
            if sig(k-1) > band1 && sig(k-1) < band2
                dic = 1; dic2 = 1; dic3 = 1; %dicrotica detectada
                idic = [idic k-1];
                ndic = ndic + 1;
            end
        end
    end
end

```

```

        break;
    end
end
end
end
if dic == 0 %no se encontró dicrotica de derivada cero
    for k = imax(j)+10:i_min-10
        min_dsig = min(dsig(imax(j)+10:i_min-10));
        if dsig(k-1) >= 0.35*min_dsig %criterio cambio derivada considerable
            if sig(k-1) > band1 && sig(k-1) < band2
                dic = 1; dic2 = 1; dic3 = 1; %dicrotica detectada
                idic = [idic k-1];
                ndic = ndic + 1;
                break;
            end
        end
    end
end
elseif dic2 == 0
    for k = imax(j)+10:i_min-10
        if dsig(k-1) >= 0.55*min_dsig %criterio cambio derivada minimo
            idic = [idic k-1];
            ndic = ndic + 1;
            dic = 1; dic2 = 1; dic3 = 1;
            break;
        end
    end
end
elseif dic3 == 0
    for k = imax(j)+10:i_min-10
        if sig(k-1) <= mn+0.5*(mx-mn) %criterio no deteccion
            idic = [idic k-1];
            ndic = ndic + 1;
            dic = 1; dic2 = 1; dic3 = 1;
            break;
        end
    end
end
end
end
end
end

```

▪ *copulse.m*

En la función *copulse()* se aplican las ecuaciones 3-13 y 3-12 mostradas en la sección 3.2, con ellas se calculan respectivamente las salidas *co_pulse2* y *co_pulse1* de la función.

```
function [co_pulse2,co_pulse1] = copulse(Pd,Ps,ki,k1,k2)
```

```

km=(ki+k1+k2)/(0.4+0.67+0.27);
co_pulse2= 500*ki.*((Ps-Pd)/(ki+k1+k2).^3);
co_pulse1= 17*(Ps-Pd)/(ki.^2);

```

▪ *esCOvar.m*

En la función *esCOvar()* se utiliza para estimar el gasto cardiaco (CO) por el metodo de PWTT usando la ecuación 3-8 mostrada en la sección 3.1.

```
function [beta, k_esCO, SV] = esCOvar(pwtt, Ps, Pd, bsa, ibeta)
%Estimación de CO por tiempo de transito de onda de pulso
```

```

if ibeta == 0
    beta = (Ps-Pd)+0.30*pwtt;
    k_esCO = NaN;

```

```
    SV = NaN;
else
    beta = ibeta;
    k_esCO = 0.59*bsa+0.31;
    SV = k_esCO .* (-0.3.*pwtt + ibeta);
end
```